NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SYSTEMS ENGINEERING AND PROJECT MANAGEMENT FOR PRODUCT DEVELOPMENT: OPTIMIZING THEIR WORKING INTERFACES

by

Mariela I. Santiago
September 2013

Thesis Advisor:	John Osmundson
Second Reader:	Gary Langford

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2013	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE SYSTEMS ENGINEERING AND PROJECT MANAGEMENT FOR PRODUCT DEVELOPMENT: OPTIMIZING THEIR WORKING INTERFACES	**5. FUNDING NUMBERS**
6. AUTHOR(S) Mariela I. Santiago	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. government. IRB protocol number ___N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

This work discusses common issues that occur from the inadequate integration of systems engineering into the project management process. In so doing, this work is shaped by the following questions: What are the most common conflicts between Program Management and Systems Engineering during product development? Where in the product development cycle do conflicts occur? How can the conflicts be mitigated?

This work identified three main conflicts within the product development process of the four case studies, the Hubble telescope, the Mars Polar Lander, the Demonstration of Autonomous Rendezvous Technology (Program, and the Constellation program. The three main problems are insufficient systems engineering in the product development process, insufficient budget and tight schedule, and inadequate risk management. These three issues eventually led to the mishaps and failures of the case studies examined in this thesis.

This work proposes that, in order to mitigate conflicts in the integration of project management and systems engineering, systems engineers and project management should be able to have a common language, understand each other's objectives, and understand how these objectives benefit both the product and the project. Therefore, its recommendations are that systems engineers be trained in project management and project managers be trained in systems engineering and that this training should include risk management. In this case, risk management is the common language between systems engineering and project management.

14. SUBJECT TERMS systems engineering, project management, product development, risk management, risk analysis, systems integration, systems interfaces,	**15. NUMBER OF PAGES** 97
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

SYSTEMS ENGINEERING AND PROJECT MANAGEMENT FOR PRODUCT DEVELOPMENT: OPTIMIZING THEIR WORKING INTERFACES

Mariela I. Santiago
Engineer, Naval Undersea Warfare Center Division Newport, Rhode Island
B.S., University of Puerto Rico Mayagüez, 1997

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author: Mariela I. Santiago

Approved by: John Osmundson, PhD
Thesis Advisor

Gary Langford, PhD
Second Reader

Clifford Whitcomb, PhD
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This work discusses common issues that occur from the inadequate integration of systems engineering into the project management process. In so doing, this work is shaped by the following questions: What are the most common conflicts between Program Management and Systems Engineering during product development? Where in the product development cycle do conflicts occur? How can the conflicts be mitigated? This work identified three main conflicts within the product development process of the four case studies, the Hubble telescope, the Mars Polar Lander, the Demonstration of Autonomous Rendezvous Technology Program, and the Constellation program. The three main problems are insufficient systems engineering in the product development process, insufficient budget and tight schedule, and inadequate risk management. These three problems eventually led to the mishaps and failures of the case studies examined in this thesis.

This work proposes that in order to mitigate conflicts in the integration of project management and systems engineering, systems engineers and project management should be able to have a common language, understand each other's objectives, and understand how these objectives benefit both the product and the project. Therefore, its recommendations are that systems engineers be trained in project management and project managers be trained in systems engineering, and that this training should include risk management. In this case, risk management is the common language between systems engineering and project management.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AVGS	advanced video guidance sensor
CDR	Critical Design Review
CONOPS	Concept of Operations
CRs	Continuing Resolution
Cx IOC	Constellation initial operating capability
DART	Demonstration Autonomous Rendezvous Technology
DS2	Deep Space 2
ESAS	Exploration System Architecture Study
GAO	United States Government Accountability Office
HDOS	Hughes Danbury Optical Systems
HLR	human lunar return
HST	Hubble Space Telescope
IC	initial capability
INCOSE	International Council of Systems Engineering
IOC	initial operating capability
ISS	International Space Station
JSC	Johnson Space Center
JPL	Jet Propulsion Laboratory
LC	lunar capability
LMSC	Lockheed Missiles and Space Company, Inc.
MIB	Mishap Investigation Board
mph	miles per hour
MPL	Mars Polar Lander
MUBLCOM	Multiple Paths, Beyond-Line-of-the-Site Communications
NASA	National Aeronautics and Space Administration
NGLT	Next Generation Launch Technology
OSC	Orbital Science Corporation
OSP	Orbital Space Plane
OTA	Optical Telescope Assembly
PBS	President’s budget submittal

P-E	Perkin-Elmer
PDR	Preliminary Design Review
PMR	program’s manager recommendation
QA	quality assurance
RNC	reflective null corrector
RvNC	refractive null corrector
2nd GRLV	2ND Generation Reusable Launch Vehicle

EXECUTIVE SUMMARY

Ulrich and Eppinger (2012, p. 2) define product development as “a set of activities beginning with the perception of market opportunity and ending in the production, sale, and delivery of the product.” In developing a product, project management and systems engineering converge to satisfy both the business process and the product process.

A Guide to the Project Management Body of Knowledge (Project Management Institute, Inc., 2013, p. 6) defines project management as “the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements”.

The *International Council of Systems Engineering (INCOSE) Systems Engineering Handbook* (v3.2) defines systems engineering as:

> …an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. (INCOSE, 2010, p. 7)

Project management focuses on the tasks required to support the development of the product with emphasis on schedule, budget, and performance. Systems engineering focuses on the technical aspects related to meeting the customer’s needs through the design and development of a solution or product. Project management is concerned with managing budgets and schedules while systems engineering is concerned with developing products and systems.

Since project management drives the project process and systems engineering deals with the product process, the project manager and the systems engineer within a project should work closely to guarantee the successful outcome of the project and the successful development of the right product with the desired performance. Success with project and product does not always happen, as will be exemplified by the discussion of the examples examined in the body of the paper, the Hubble telescope, the Mars Polar Lander, the Demonstration of Autonomous Rendezvous Technology (DART) Program, and the Constellation program.

Budget and schedule drive projects while milestones drive the systems engineering process and the product development process. As such, conflicts can arise between the project management and the systems engineering objectives.

In organizations where project management guides the project process and systems engineering guides the product process, it is imperative that these two processes work in congruence. Failure to do this may result in cost and schedule overruns and in poor product performance. The case studies discussed in this work will provide examples of cost and schedule overruns and poor product performance.

This work discusses common issues that occur from the inadequate integration of systems engineering into the project management process. This work also identifies where in the product development cycle the conflicts occur and ways to mitigate the issues.

The case studies discussed in this work exemplify programs that encountered technical issues or mishaps due to either inadequate integration of systems engineering with the project management process.

Three main conflicts within the product development process have been identified by this work: insufficient systems engineering in the product development process, insufficient budget and tight schedule, and inadequate risk management. These three situations eventually led to the mishaps and failures of the case studies presented.

This work concludes that the issues mentioned above result from either starting systems engineering late in the process or as insufficient application of systems engineering processes in the project as a cost reduction effort.

As presented through the different case studies, the investigation boards assigned to the different programs identified issues throughout the product development process. However, in all the cases, it can be stated that failure to establish an adequate systems engineering process in the early planning stages of the product development process resulted in issues in the later stages of the process. Examples of the issues on later stages are inability to conduct verification and validation efforts due to poorly elicited requirements or the lack of documentation of the requirements elicitation, the design process, analyses of alternatives, and the validation and verification processes.

This work proposes that, in order to mitigate conflicts in the integration of project management and systems engineering, systems engineers and project management should strive to have a common language, work to be able to understand each other’s objectives, and try to understand how these objectives benefit both the product and the project.

This understanding and common language, this work proposes, could be achieved through the effective training of project managers in systems engineering and systems engineers in project managers. This training should include risk management. Risk management could be the common language between systems engineering and project management. This understanding and common language could result in better allocation of resources, improved budget and schedule management, and better control of project scope.

LIST OF REFERENCES

International Council of Systems Engineering. (2011). *INCOSE systems engineering handbook v. 3.2.2.* San Diego: International Council of Systems Engineering.

Project Management Institute, Inc. (2013). *A guide to the project management body of knowledge (PMBOK(R) GUIDE).* Newtown Square, PA: Project Management Institute, Inc.

Ulrich, K., & Eppinger, S. (2012). *Product design and development* (5th ed.). New York: McGraw-Hill/Irwin.

ACKNOWLEDGMENTS

I would like to thank the Naval Undersea Warfare Center Division Newport management for the trusting support that helped made this project a reality:

Mr. Michael Demetri, Mr. Mark Rodrigues, Mr. Carlos Galliano, Mr. Tim Levandowski, Mr. James Broadmeadow, Mr. Don Aker, and the staff at the Office of Strategic Management.

Special thanks to Dr. Osmundson and Dr. Langford for contributing their expertise to this work and for their patience and guidance.

I would like to acknowledge Professor Mary Vizzini and Catherine Grant for assistance in reviewing this work's structure, language, and organization.

Thanks to Heather Hahn for keeping this process rolling and working so diligently and efficiently.

Special thanks to my parents and siblings, for shaping me into who I am and to my grandma, Mami Iris, for pretty much EVERYTHING and for giving me the power to dream.

I dedicate this work to:

- *my loving husband, Luis Gonzalez, for his unyielding love and support for the last 14 ½ years*
- *my dear daughter, Paula Gonzalez, for her love and patience while Mommy juggled school, work, and home*
- *God, the creator of the Universe, for free will, for the talents and the graces provided.*

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

In developing a product, ideally project management and systems engineering converge to satisfy both the business process and the product process. Project management focuses on the tasks required to support the development of the product with emphasis on schedule, budget, and performance. Systems engineering focuses on the technical aspects related to meeting the customer's needs through the design and development of a solution or product.

Project management is concerned with managing budgets and schedules while systems engineering is concerned with developing products and systems. Because of these differing concerns, conflicts can arise between the project management and the systems engineering objectives.

Budget and schedule drive projects, while milestones drive the systems engineering process and the product development process. Thus, conflicts can arise between the project management and the systems engineering objectives.

This work discusses some of these conflicts through case studies, specifically, the Hubble telescope, the Mars Polar Lander, the Demonstration of Autonomous Rendezvous Technology (DART) Program, and the Constellation program, and identifies where in the product process they happen, and discusses ways in which these can be resolved or prevented.

B. PRODUCT DEVELOPMENT

Ulrich and Eppinger (2012, p. 2) define product development as "a set of activities beginning with the perception of market opportunity and ending in the production, sale, and delivery of the product." Throughout this work, the product development processes is represented by the Ulrich's and Eppinger's generic model shown in Figure 1.

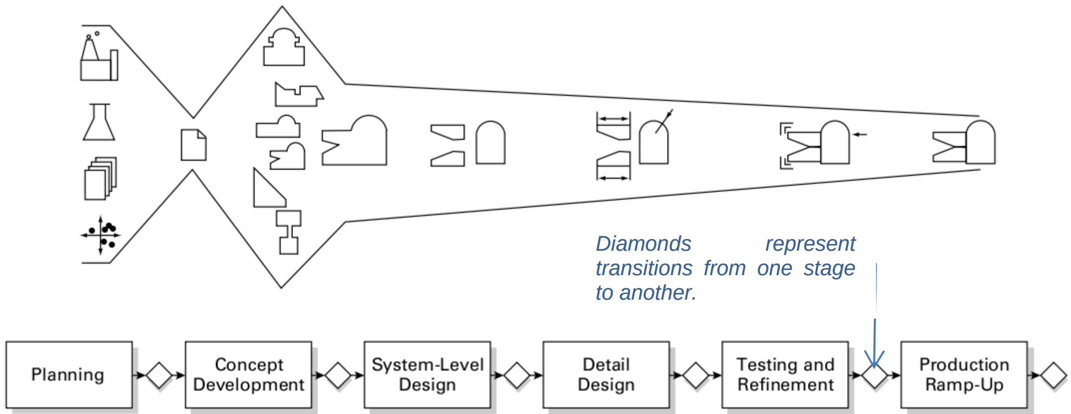


Figure 1. Generic product development process model (After Ulrich & Eppinger, 2012)

Ulrich and Eppinger's model outlines the following steps:

- Planning: This phase involves the development of the approach proposed to achieve the desired product. The planning phase identifies things like the customer, product functionalities and top-level requirements, and schedule and cost restrictions.
- Concept development: During this phase, the team identifies the customer's needs, develops concepts, and establishes requirements that are more detailed. A concept may be down selected or various concepts may be "selected for further development and testing"
- System-level design: This phase involves the functional decomposition of the product into systems architecture.
- Detail design: During detail design, the project team will determine detail specifications for the product (e.g., "geometry", "tolerances") as well as the required manufacturing, fabrication, and assembly processes. Documentation is an important part of this phase as it will track the history of the product development and manufacturing and will trace to future stages.
- Testing and refinement: The testing and refinement phase involves the evaluation of the product to ensure it meets pertinent requirements. The main objective is to validate and verify that the product meets the intended need and that its "performance and reliability" are acceptable. The testing and refinement phase allows improvements to the product, usually through the use of prototypes, prior to the start-up of the manufacturing phase.

- Production ramp-up: The production ramp-up phase builds a few of the product at a low production rate, establishing the required manufacturing system while allowing for any needed improvements to the product or the process itself.

Product development takes place within an organization usually under a program or project plan. Programs and projects are managed using project management techniques. As such, the project schedule usually drives the product process.

The next section describes project management and its relationship with product development.

C. WHAT IS PROJECT MANAGEMENT?

A Guide to the Project Management Body of Knowledge provides the following definition:

> Project: a temporary endeavor undertaken to create a unique product, service, or result. The temporary nature of projects indicates a definite beginning and end. The end is reached when the project's objectives have been achieved or when the project is terminated because its objectives will not or cannot be met, or when the need for the project no longer exists. (Project Management Institute, Inc., 2013, p. 3)
>
> Project Management: the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements. (Project Management Institute, Inc., 2013, p. 6)

The PMBOK states that project management requires a set of management processes to ensure project goals are accomplished. The book groups this processes into five different process groups:

- The Initiating Process Group gives way to the start of a project. It establishes the view and mission.
- The Planning Process Group provides the roadmap for the view and mission from the Initiating Process Group.
- Executing Process Group implements the plans established to achieve the project goals.

- Monitoring and Controlling Process Group ensures the project plan carries on as expected and makes adjustments down the line as needed to ensure adaptability to changes.
- Closing Process Group "finalizes activities across all Process Groups to formally close the project or phase." (Project Management Institute, Inc., 2013, p. 39)

Figure 2 depicts how each of these processes start and end within the project process.

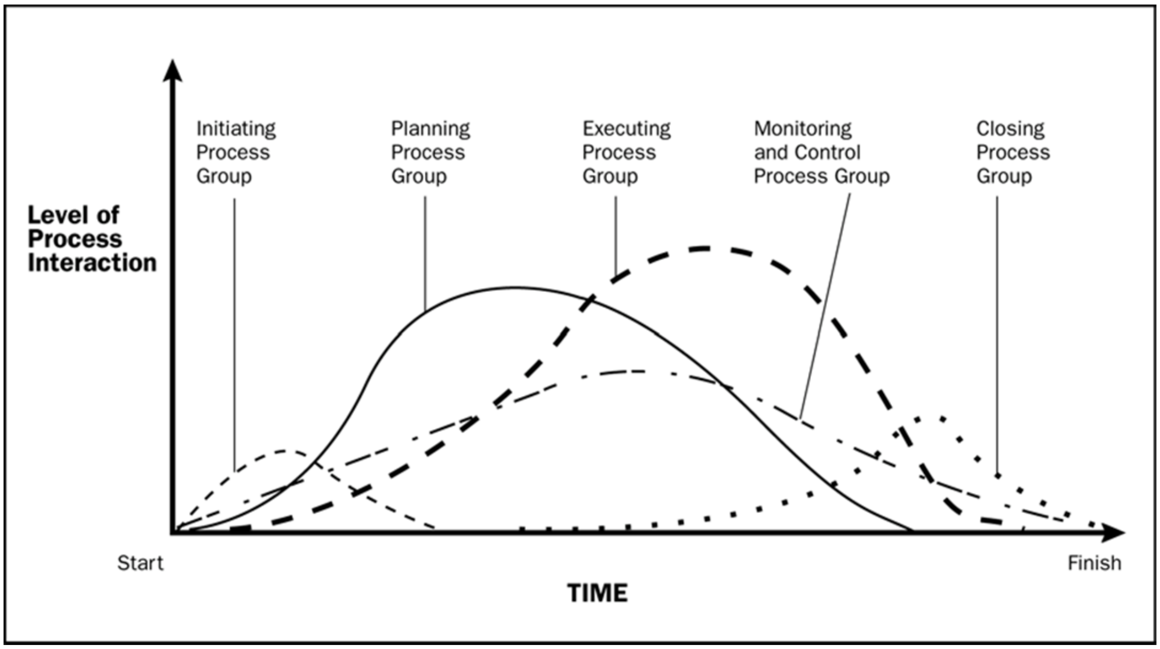


Figure 2. Project management process groups interact in a phase or project (From Project Management Institute, Inc., 2013)

Project management's objective is to deliver the product on time and within schedule. The process groups manage the project effort to ensure successful completion of the objective.

The PMBOK, however, does not address the product processes (i.e., processes required to ensure the project's deliverable meets the customer's needs and performs in a reliable manner and within the established specifications).

Systems engineering, on the other hand, is about ensuring adequate identification of customer's needs and a product that meets all established requirements. The next section discusses systems engineering.

D. SYSTEMS ENGINEERING

INCOSE's *Systems Engineering Handbook* (v3.2) defines systems engineering as:

> …an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. (INCOSE, 2010, p. 7)

Figure 3 shows the DoD 2009 systems engineering process. This figure is presented as an example of a commonly used systems engineering process model.

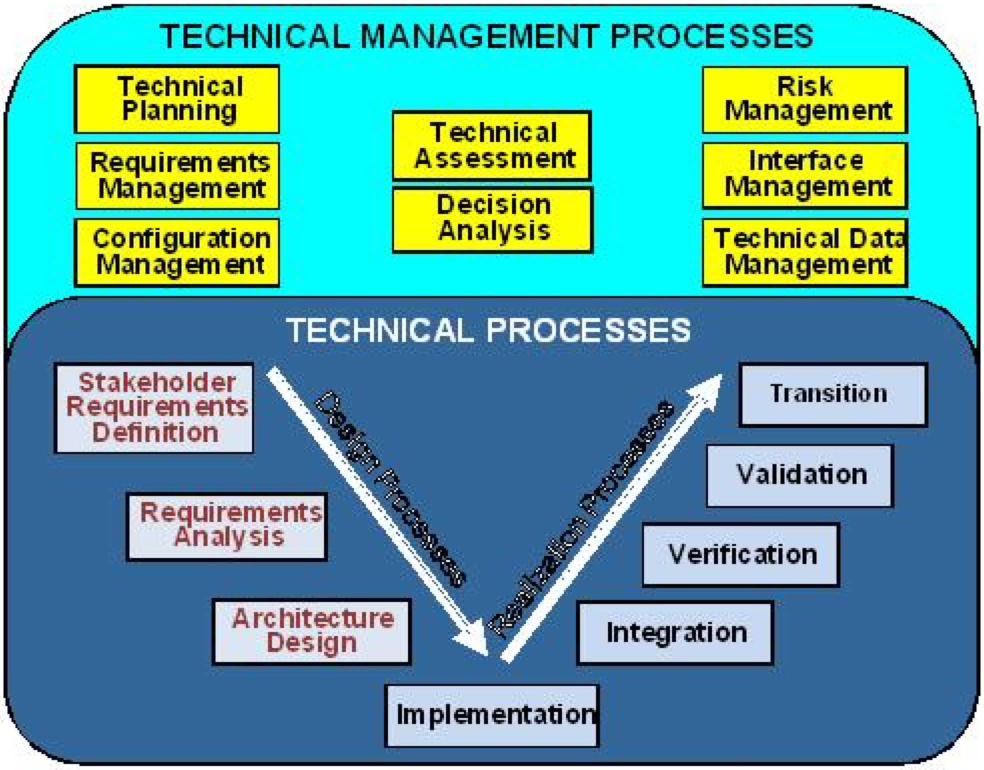


Figure 3. DoD systems engineering process model (From Department of Defense, 2011)

Although systems engineering processes may differ somewhat from organization to organization, they all have the following basic steps: stakeholder analysis, identification of customer's need or problem, functional decomposition and requirements analysis, detail design and systems architecture, test and evaluation (verification and validation), and implementation.

- Stakeholder's analysis involves the identification of important players in the development of the product. Stakeholder's may be product's users, project's sponsors, developers, designers, manufacturers; any entity that may in some way have an input into the requirements and standards that will guide the product development.
- Identification of customer's need or problem will ensure that the team is solving the right problem so that the adequate product is developed. This process requires extensive communications with the stakeholders and the representation of the problem statement in a language that identifies specific goals.

- Functional decomposition and requirements analysis break down the problem statement into achievable and measurable goals. Functional decomposition identifies the functions that the product shall perform and assigns performance measures and specifications. In addition to performance requirements, attributes such as weight, size, appearance, and human factors are also part of the product requirements.
- Detail design and systems architecture translates the functional decomposition into system architecture. The Systems engineering team will assign requirements and specifications to the subsystems and components of the system. During this stage, the Systems engineering will pay special attention to the decomposition of requirements from top-level systems requirements to sub-systems requirements to component level requirements and specifications. Another paramount task during this stage is tracking system's interface requirements. Components that may work adequately by themselves may malfunction or inadequately interface when integrated as a subsystem or a system.
- Test and evaluation (verification and validation) helps ensure that the team built the right product and that they built it right. The results from this work can only be as good as the requirements and specifications from which it derives its evaluation methods. This work highlights the importance of adequate definition of requirements and suitable functional decomposition. Inadequate requirements definition will lead to a product that may not meet the customer's needs or a defectively built product.
- Implementation brings the product to the customer. Data is usually still collected to investigate the capability of the product to meet the customer's need. The systems engineering team will collect data from the customer; useful for any further developments of the product.

As shown by the steps above, in a product development environment systems engineering deals with the product process. Early incorporation of systems engineering process into the project helps with problem definition and product functionality and eventually diminishes the probability of design changes later in the process. The later in the process changes in design are made, the costlier it is for the project (see Figure 4).

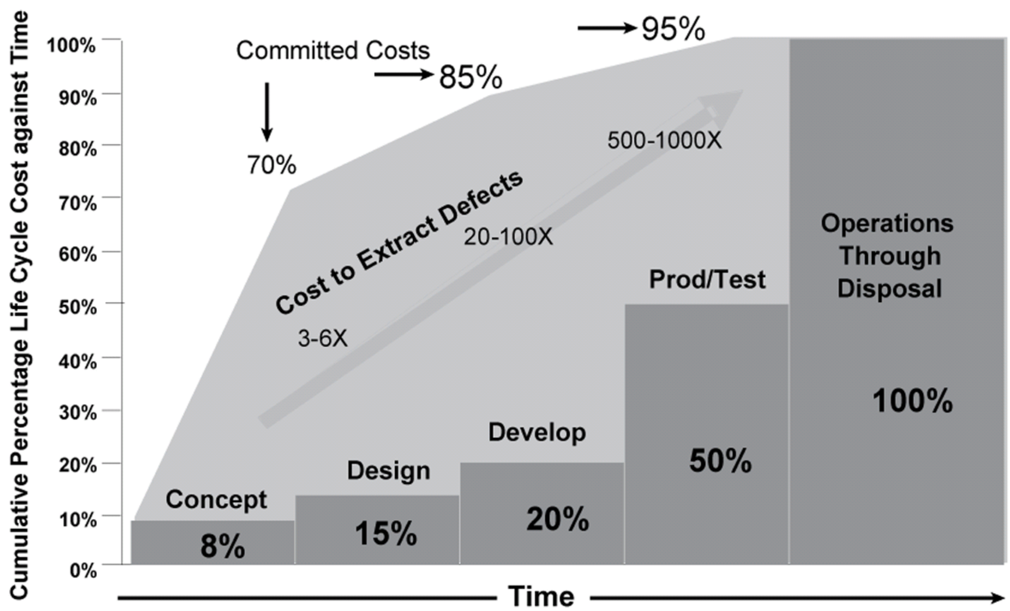


Figure 4. Cumulative percentage life cycle cost against time (From INCOSE, 2011)

Given that project management drives the project process and systems engineering deals with the product process, it is only logical that these two disciplines should work closely to guarantee the success of the project and successful development of the right product. Success with project and product does not always happen, as will be exemplified by the discussion in this work.

E. RESEARCH QUESTIONS

In organizations where project management guides the project process and systems engineering guides the product process, it is imperative that these two processes work in congruence. Failure to do this may result in cost and schedule overruns and in poor product performance.

This work discusses common issues that occur from the inadequate integration of systems engineering into the project management process. As such, this work researches the following questions:

- What are the most common conflicts between Program Management and Systems Engineering during product development?

- Where in the product development cycle do conflicts occur?
- How can the conflicts be mitigated?

F. BENEFITS OF STUDY

It is hoped that the results from this study will provide useful guidance and information on how to improve product development.

G. SCOPE AND APPROACH

1. Scope

This work discusses issues with inadequate systems engineering integration with the project management process using various representative NASA programs.

2. Approach

This work starts by presenting fundamental concepts of product development, project management, and systems engineering. It continues on to discuss various National Aeronautics and Space Administration (NASA) programs that encountered issues or mishaps due to either inadequate integration of systems engineering with the project management process or for missing systems engineering steps within the project process. Each case is analyzed separately. Where applicable, supporting literature review on product development, project management, or systems engineering is discussed.

H. ORGANIZATION OF STUDY

This work is organized in seven different chapters. Chapters II through V discuss the Hubble telescope, the Mars Polar Lander, the Demonstration of Autonomous Rendezvous Technology (DART) Program, and the Constellation program case studies respectively. Chapter VI contains a discussion of the case studies and similar findings in literature review. Chapter VII includes the final conclusions and recommendations.

THIS PAGE INTENTIONALLY LEFT BLANK

II. CASE STUDY 1: HUBBLE TELESCOPE

A. INTRODUCTION

This chapter discusses NASA's Hubble telescope program. A background on the program in presented, followed by an account of the systems failure.

B. BACKGROUND

Scientists and engineers developed the Hubble Space Telescope (HST) (see Figure 5), with the ultimate objective of deepening our understanding of the universe. A space telescope would provide images like no other telescope before. The images would be free of the limitations imposed by the Earth's atmosphere.

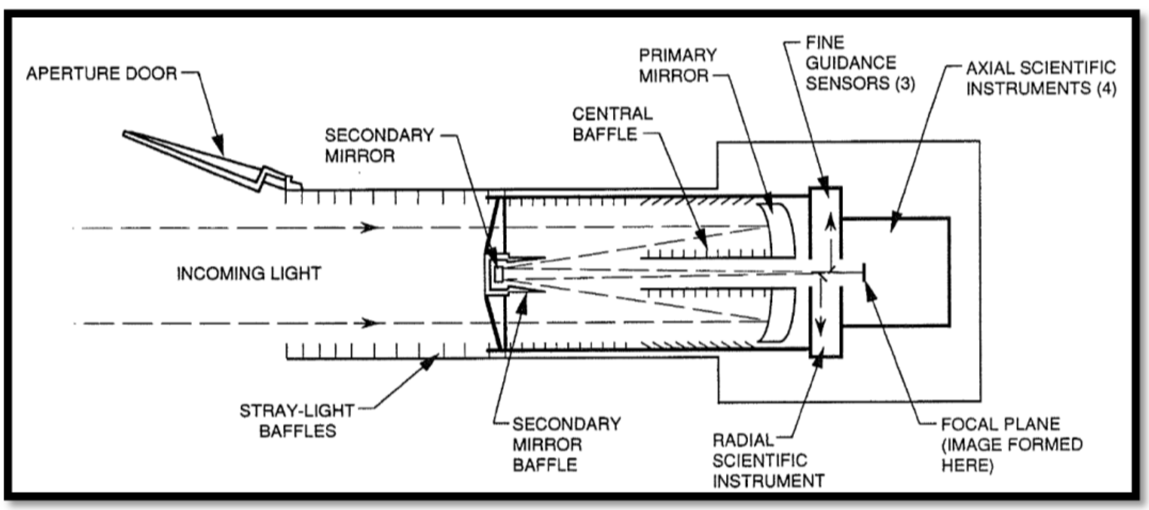


Figure 5. Optical telescope assembly. "The Optical Telescope Assembly has a 2.4-m Ritchey-Chretien telescope with a focal ration of f/24. The optical range of the Hubble Space Telescope extends from 1,100 to 11,000 angstroms, and the performance quality in the ultraviolet is unique." (From National Aeronautics and Space Administration, 1990, pp. 2-1–2-2)

The work for the HST was divided among different organizations. Figure 6 shows the breakdown of responsibilities associated with the development and fabrication of the HST. As can been seen from the figure, Perkin-Elmer Corporation (P-E) was responsible for the design, build, and assembly of the optical telescope assembly (OTA). Lockheed Missiles and Space Company, Inc. (LMSC) was responsible for the development of the support systems module, full systems engineering and systems integration, as well as the supervision of other subcontractors.

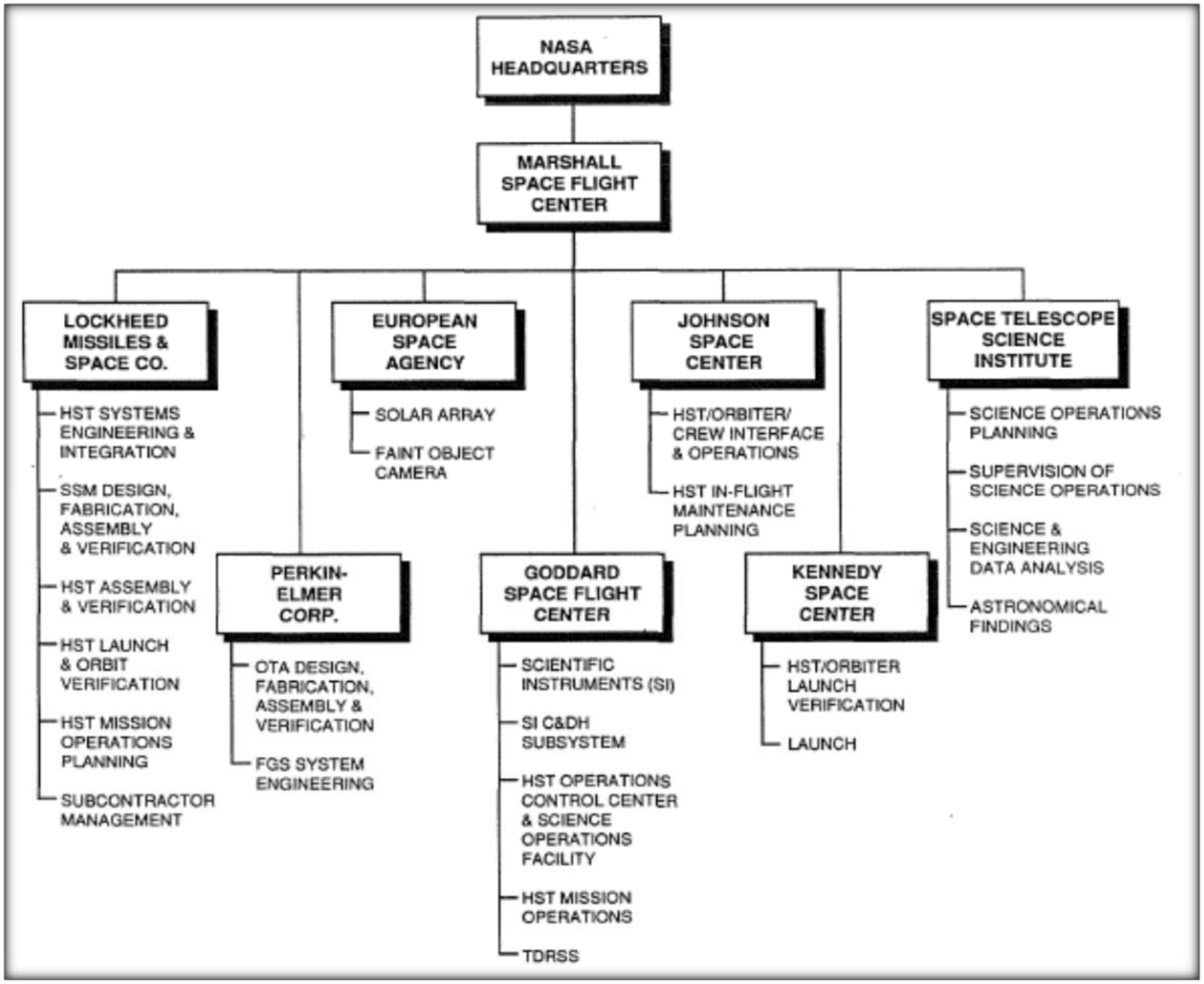


Figure 6. Breakdown of responsibilities for HST development (From (National Aeronautics and Space Administration, 1990)

C. THE MISHAP

The "Hubble Space Telescope Optical Systems Failure Report" describes the HST mishap as follows:

> The rough grinding operation for the Hubble Space Telescope began in December 1978, at the Perkin-Elmer Corporation, in Wilton, Connecticut. The mirror was then transferred to Perkin-Elmer in Danbury, Connecticut, now Hughes Danbury Optical Systems, Inc. (HDOS), where polishing was completed in April 1981, and the mirror was accepted as ready for reflective coating. The final post-coating test was made in February 1982.
>
> Approximately two months after launch, on June 21, 1990, the Hubble Space Telescope Project Manager announced that there was a major flaw in one or both of the mirrors in the Optical Telescope Assembly. (National Aeronautics and Space Administration, 1990)

In summary, a thorough investigation led by the Hubble Space Telescope Optical Systems Board of Investigation discovered that the telescope had myopic vision because it had been ground into the wrong shape. A 1 mm error in the reflective null corrector (RNC) went undetected by Perkin-Elmer (P-E) developers and their acceptance testing (National Aeronautics and Space Administration, 1990). The RNC was a newly developed set up for the HST by Perkin-Elmer. P-E considered existing techniques, refractive null correctors (RvNC), not accurate enough for the HST. Figures 7 and 8 show the set up for an RNC and an RvNC respectively.

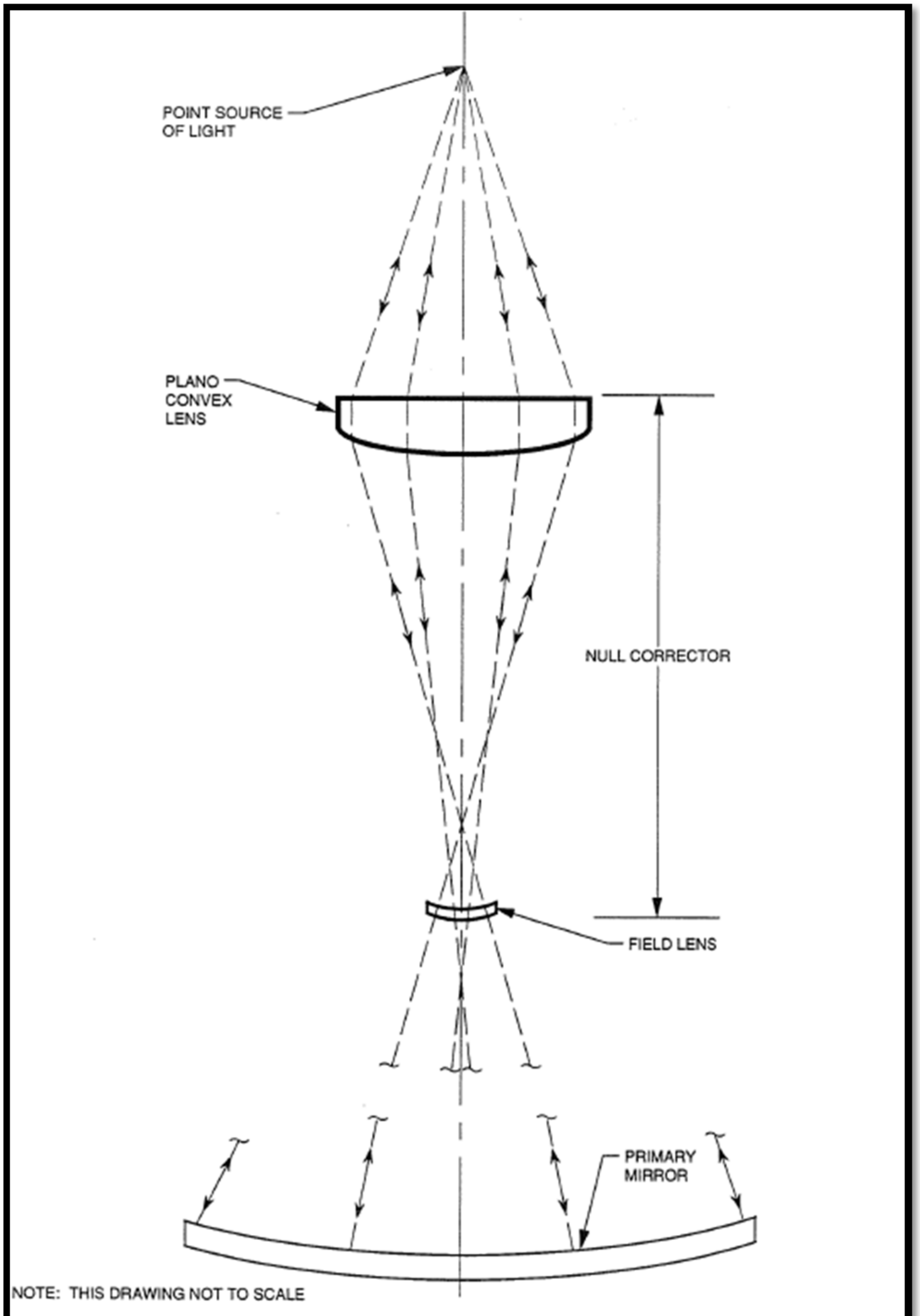


Figure 7. Two element refractive null corrector. (From National Aeronautics and Space Administration, 1990)

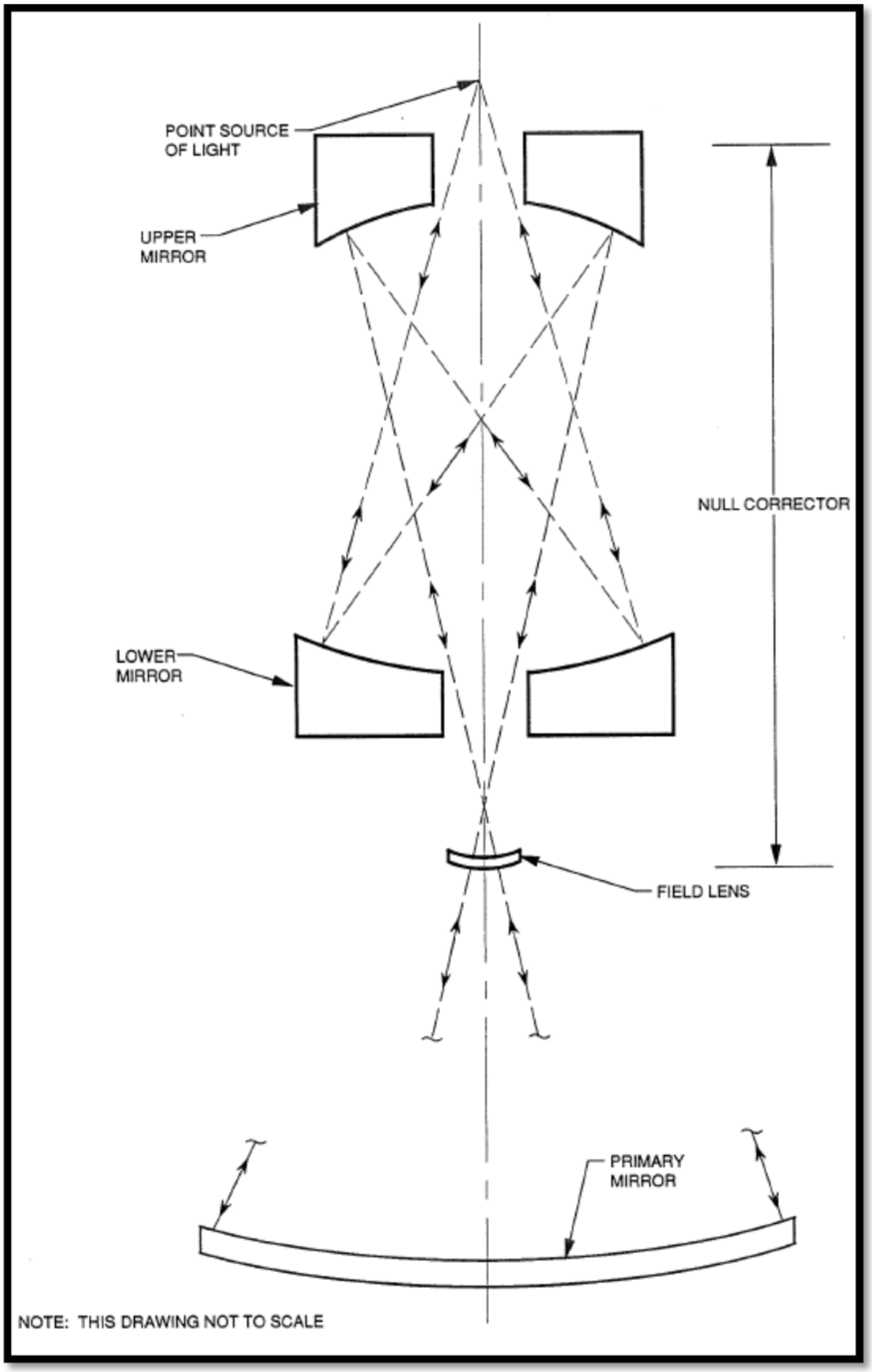


Figure 8. Reflective null corrector developed for the HST program (From (National Aeronautics and Space Administration, 1990)

D. FINDINGS AND RECOMMENDATIONS OF THE HUBBLE SPACE TELESCOPE OPTICAL SYSTEMS BOARD OF INVESTIGATION

The *Hubble Space Telescope Optical Systems Failure Report* (National Aeronautics and Space Administration, 1990) states that initial testing with the refractive null corrector showed evidence of spherical aberration on the primary mirror. P-E discarded the results, thinking there was something wrong with the refractive null corrector. No independent review or test was conducted. Tests conducted by P-E prior to launch and using the reflective null corrector indicated the mirror exceeded the required specifications. This post launch testing by P-E revealed evidence of the problems with the telescope mirror prior to launch, i.e., a manufacturing defect. The question was "why was it not identified prior to launch" (National Aeronautics and Space Administration, 1990, p.iii)? The Hubble Space Telescope Optical Systems Board of Investigation found that several issues within HST product development process led to the situation.

1. Root Causes: Quality Control and Documentation

Figure 9 show the phases within the product development process where the HST program exhibited problems.

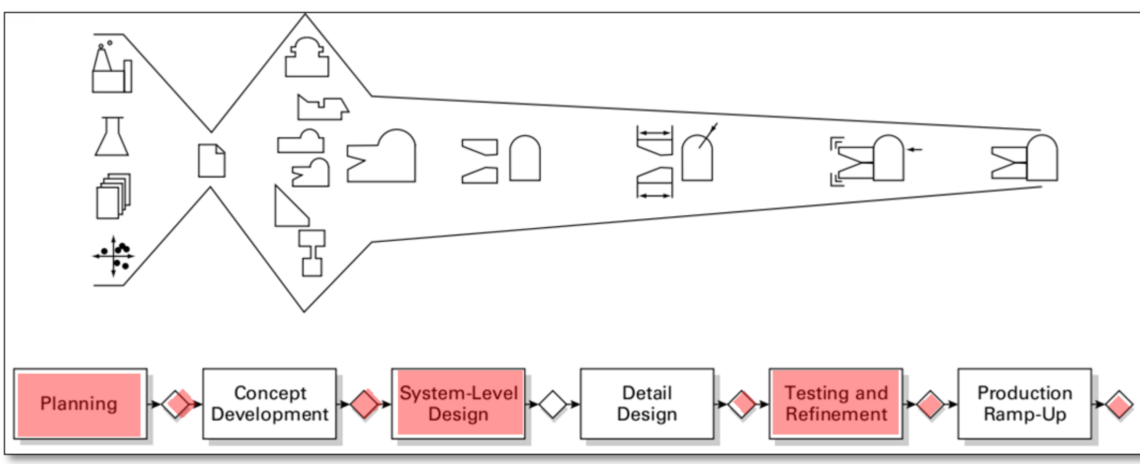


Figure 9. Phases within the HST product development process where issues were identified. (Phases highlighted in red showed issues) (After Ulrich & Eppinger, 2012)

The Board of Investigation determined that a quality assurance (QA) plan was developed for the OTA. However, the details in the plan were not clear. Specifically, there was a lack of traceability of QA requirements to specific components and to testing requirements. The QA plan did not provide enough direction and requirements so as to offer an effective evaluation of the polishing and testing processes.

In addition, the QA activity was not adequately staffed and, in apparent conflict of interests, the QA personnel reported to the OTA project manager. The result was an ineffective and limited QA team that was the subject of pressure of project budget and schedule.

The QA plan is usually establish during the planning stages of the product development process. However, this plan should be revised when transitioning from one product development process stage to the other. Specifically, entrance criteria to design reviews should include updates and status on QA processes.

2. Root Causes: Risk Management and Team Communications

The Board of Investigation concluded that the HST program did not have an adequate Risk Management process. The team failed to identify the mirror manufacturing as a high-risk undertaking and therefore did not properly implemented mitigation steps to counter any issues.

Risk management starts in the planning phases just as the QA plan. Risk management evolves with the developing project and requires constant discussion and assessment from the team. A good systems engineering analysis would have identified as a high risk that the fabrication and testing of the primary mirror did not have adequately defined QA requirements.

Another high-risk item identified by the HST Board of Investigation was the lack of interaction between the component developers. According to the HST Board “contributing to poor communications was an apparent philosophy at Marshal Space and Flight Center at the time to resolve issues at the lowest possible level and to consider problems that surfaced at reviews to be indications of bad management" (National Aeronautics and Space Administration, 1990, p.10-2). Having a systems engineering team that can identify high-risk items

throughout the project and help serve as a communication link between the different sub-subsystems teams can greatly increase the likelihood of project success.

3. Root Causes: Schedule and Cost Pressures

The Board of Investigation states that the HST management were under schedule and budget pressures.

> At one point during the fabrication cycle of the primary mirror, an urgent recommendation for independent tests to check for gross error entered the system, but was apparently not acted upon….. at the completion of mirror polishing, the final review of data for a final report was abandoned and the team reassigned as a cost-cutting measure. (National Aeronautics and Space Administration, 1990, p. 10-4)

Dealing with cost and schedule pressures is not easy. A risk mitigation plan and strict control gates (design reviews) would have helped ease the pressures on the HST program. Forsberg and Mooz (2002) state that control gates (design reviews) should pay attention to the project evolving business case and identify, if needed, ways in which the project must be adapted to meet the new business realities.

Good communication amongst team members and organizations involved, as well as excellent risk management processes will make it possible for a project to react effectively to changing needs.

E. CONCLUSIONS

Waldrop (1990, p. 735) stated that the HST blunder might have happened "due to a combination of managerial laxness and technological hubris." In reality, what the HST case shows are the dangers of failing to carry on an effective Risk Management plan and sound systems engineering practices.

Effective systems engineering should have ensured adequate communication among the subsystems and components developers and would have identified high-risk items within the program.

Most importantly, paying attention to the evolving business case and scrutinizing changes within the business case during control gates would have helped guide technical decisions in the face of schedule and budget pressures.

III. CASE STUDY 2: MARS POLAR LANDER

A. INTRODUCTION

This chapter discusses NASA's Mars Polar Lander program. A background on the program in presented followed by an account of the systems failure.

B. BACKGROUND

According to the Jet Propulsion Laboratory Special Review Board (2000), "The Mars Polar Lander (MPL), with two Deep Space 2 (DS2) probes, was launched on 3 January 1999 for arrival at Mars on 3 December 1999" (Jet Propulsion Laboratory Special Review Board, 2000, p. 4).

The objective of the MPL (Figure 10) and DS2 (Figure 11) mission was "to address the science theme:

> …volatiles and climate history on Mars, thereby directly addressing the climate-history and resource themes of the Mars Surveyor Program, while supporting the life-on-Mars theme through characterization of climate change and its evolving impact on the distribution of water. (NASA Jet Propulsion Laboratory, 1998)

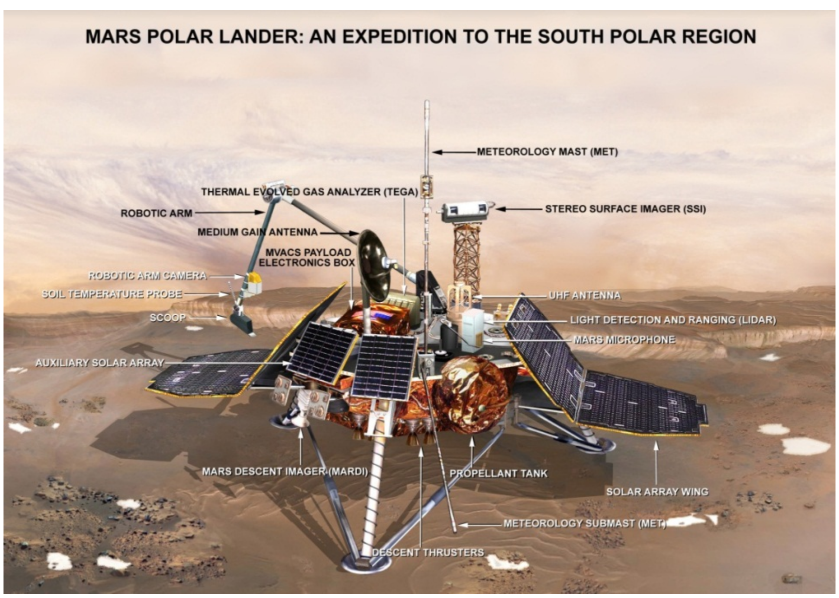


Figure 10. Mars Polar Lander (From NASA Jet Propulsion Laboratory, n.d.)

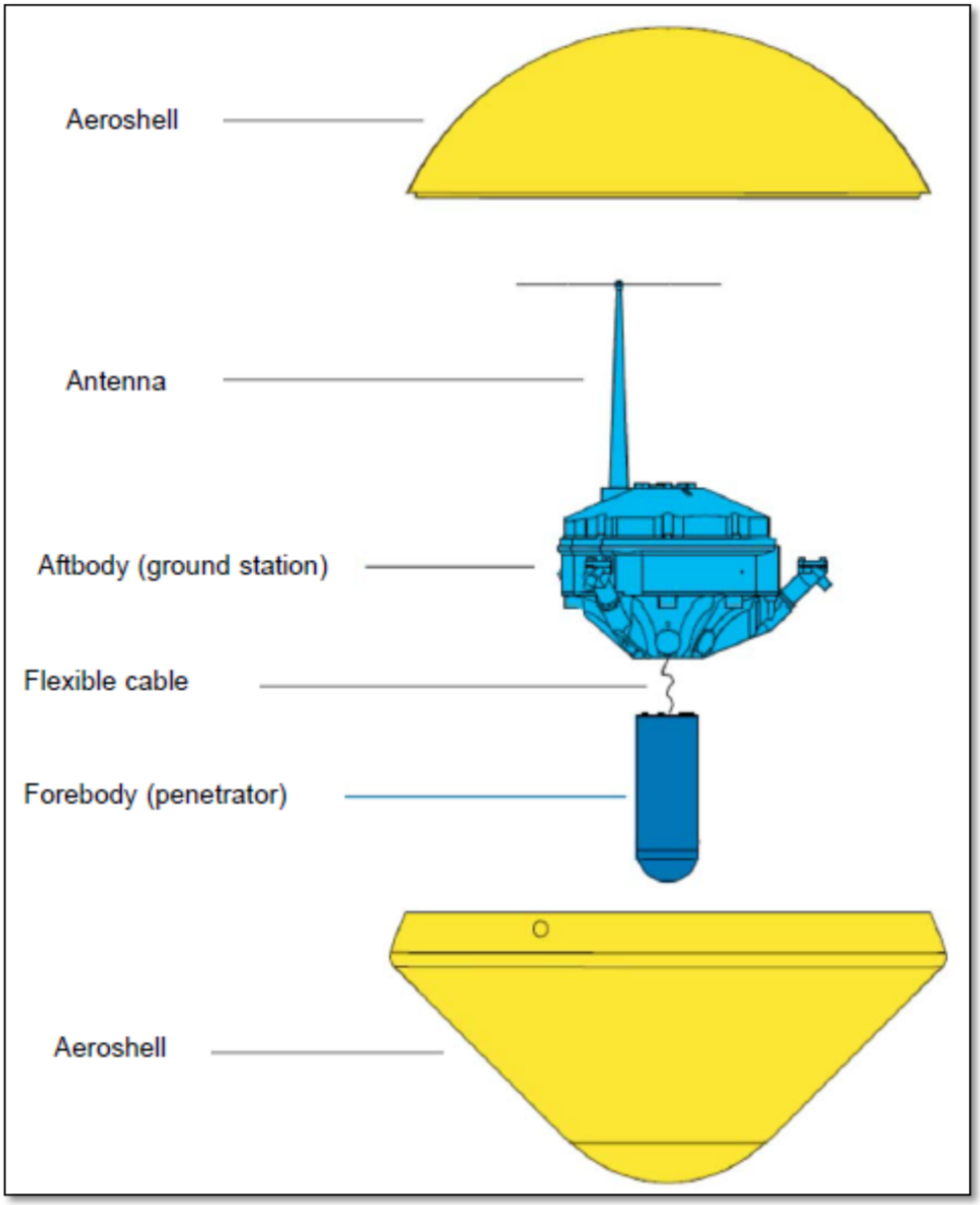


Figure 11. DS2 (From National Aeronautics and Space Administration, 1999)

1. CONOPS

The MPL DS2 mission was ambitious. In addition to the analysis and study of water on Mars, the mission would test several new technologies. The technologies would enable soil sampling, meteorology analysis, seismic monitoring, the detection of carbon dioxide and ice water within the soil, and photographing the area around the lander. Figures 12 and 13 show some of the technologies that the MPL and the DS2 were carrying.

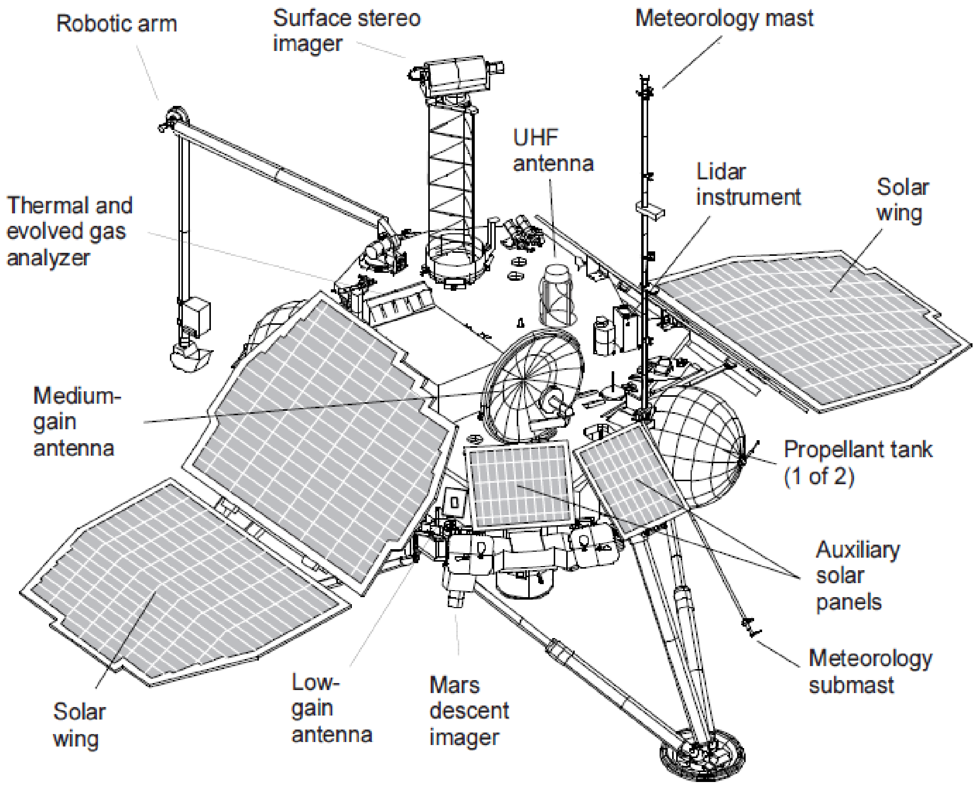


Figure 12. Instruments on board the MPL (From National Aeronautics and Space Administration, 1999)

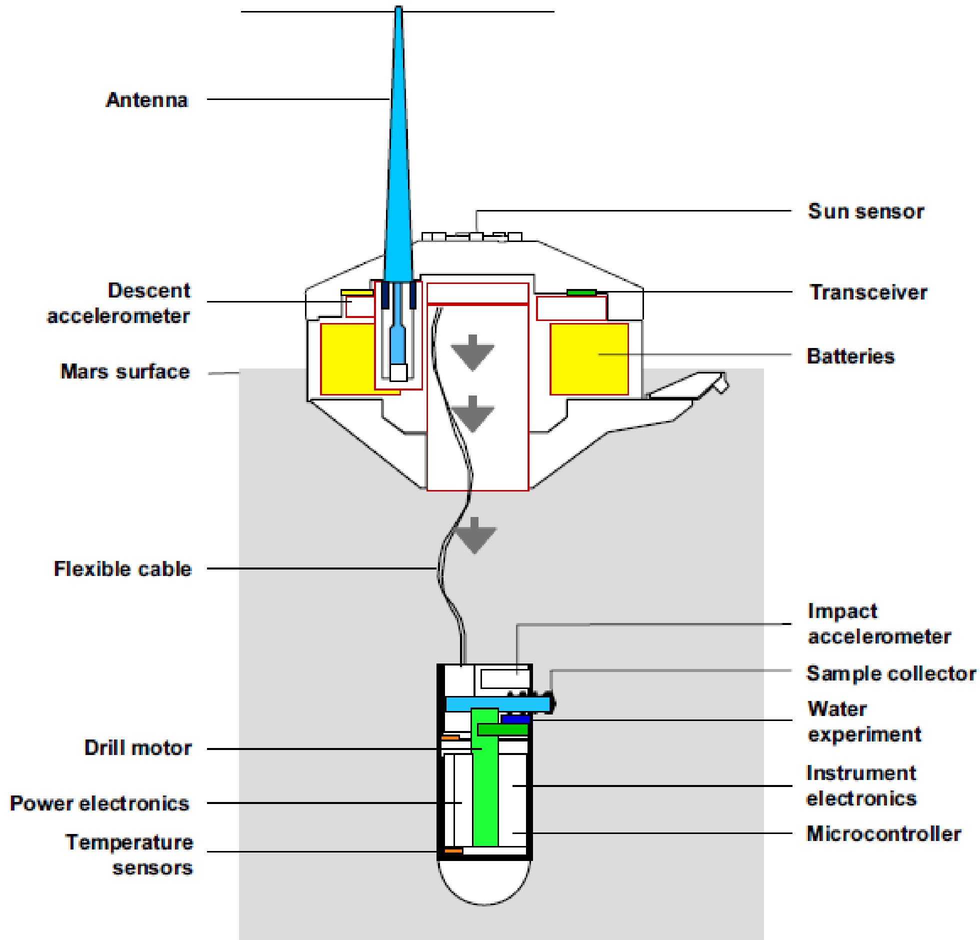


Figure 13. Instrumentation on board DS2 (From National Aeronautics and Space Administration, 1999)

The MPL left the Earth on January 3, 1999. It was expected to reach Mars on December 3, 1999. The MPL was to use retro rockets during its landing stage onto Mars. The retro rockets purpose was to decelerate the lander. The entry, descent, and landing CONOPS presented on Figure 14 is described in the Mars Polar Lander/ Deep Space 3 Press Kit (National Aeronautics and Space Administration, 1999) as follows:

- Before entering the Mars atmosphere, cruise stage would be jettisoned. It is at this point that the DS2 would separate from the lander.
- The spacecraft would be traveling at 15, 400 miles per hour (mph) when entering the Mars atmosphere.

- The parachute deployment was scheduled to happen around the point where the lander was traveling at 960 mph. At this point, the lander would jettison the heat shield.
- The deployment of the lander legs would happen at around “70 to 100 seconds before landing” followed by the landing radar start-up. (National Aeronautics & Space Administration, 1999, p. 20)
- Following “radar ground acquisition” the lander would separate from the backshell and the descent engines would start-up. These descent engines would have kept the lander in the right bearings for final touchdown. (National Aeronautics & Space Administration, 1999, p. 20)
- “At an altitude of 40 feet or a velocity of 5.4 mph the lander would drop straight down at a constant speed. The descent engines would be turned off when touchdown is detected by sensors in the footpads.” This last step, this work concludes, would later prove to be at the center of the MPL loss discussion. (National Aeronautics & Space Administration, 1999, p. 22)

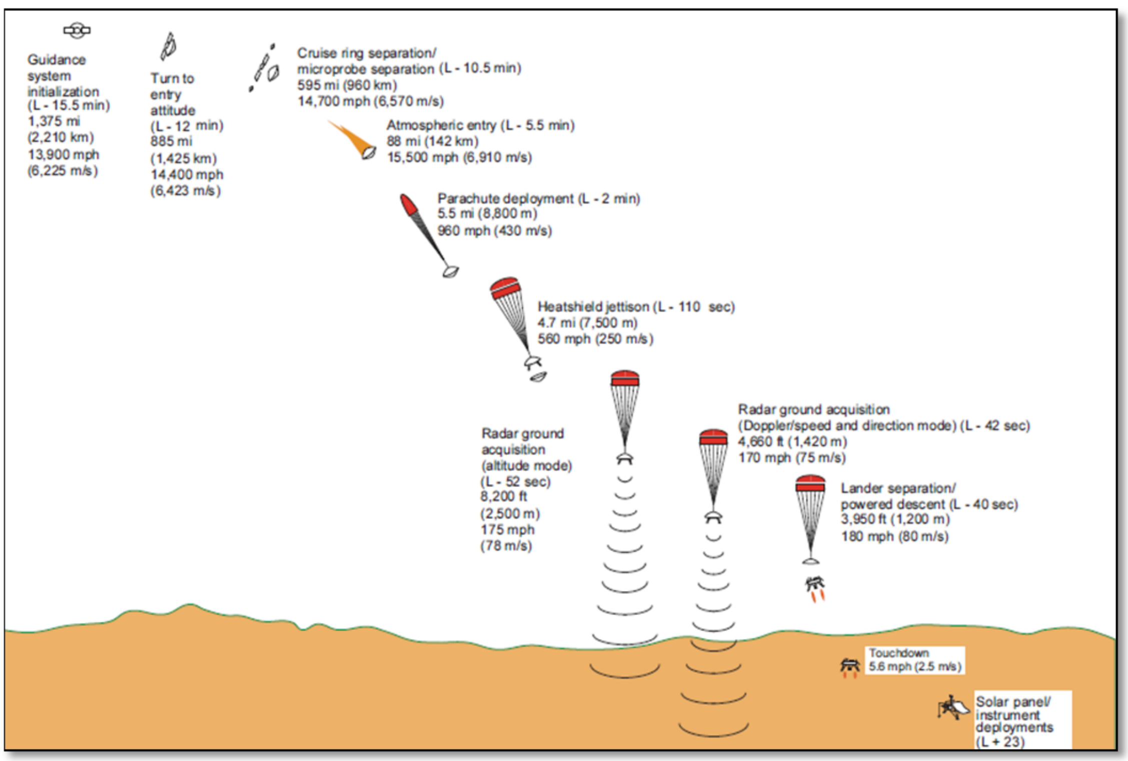


Figure 14. Entry, descend, and landing sequence for the MPL and DS2 (From National Aeronautics and Space Administration, 1999)

Figure 15 shows the lander flight systems and the different jettison stages.

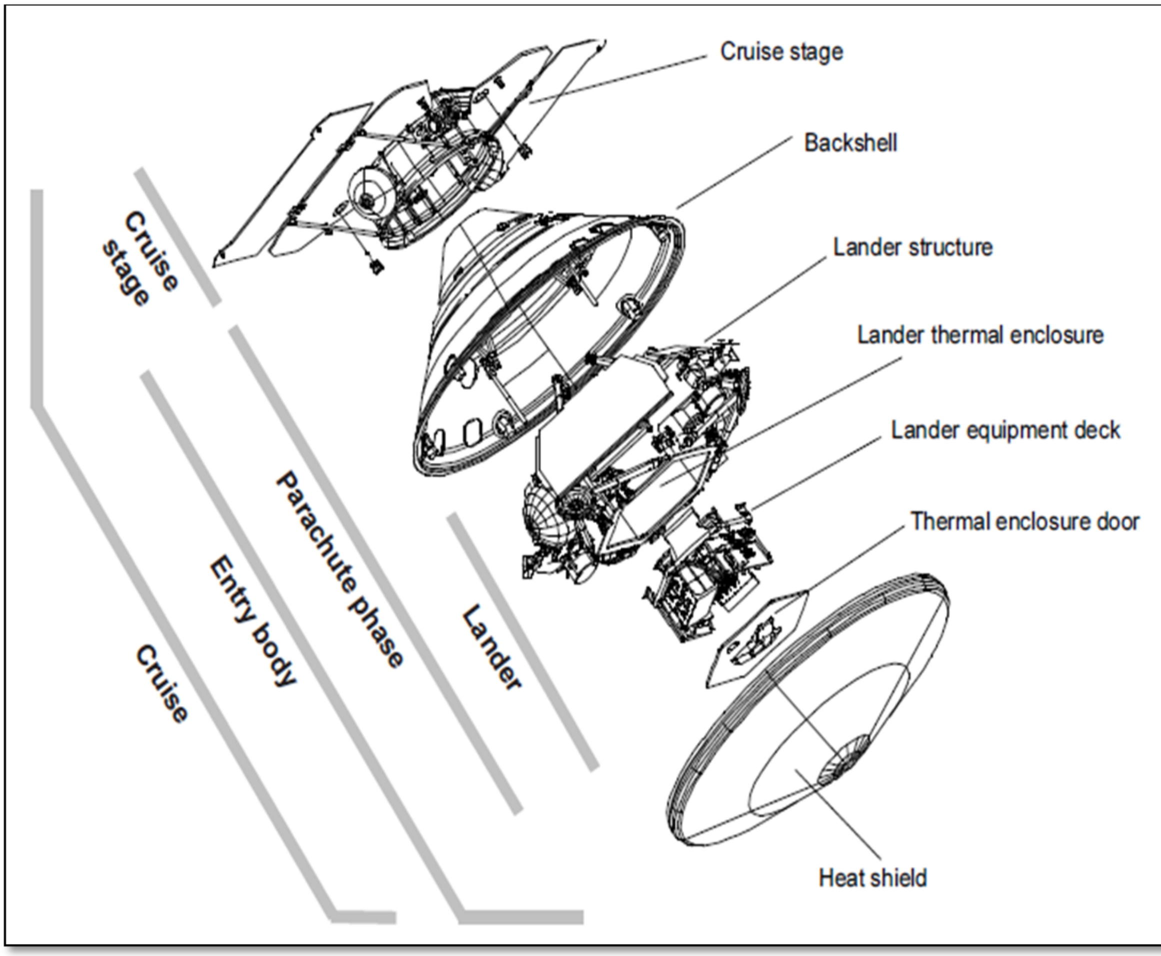


Figure 15. MPL flight system (From National Aeronautics and Space Administration, 1999)

C. THE MISHAP

MPL reached Mars as expected on December 3, 1999. Around half an hour after touchdown, the MPL team should have started receiving communications. It never happened. MPL never communicated. Late January 2000 dates the last attempts from the team to communicate with MPL. Attempts to gather visual information using the Mars Orbiter Camera also proved unsuccessful.

The Jet Propulsion Laboratory (JPL) Review Board was established on December 16, 1999 to determine the root cause behind the failure of the mission. The JPL Review Board identified several possible technical reasons as the cause behind the failure. The most likely cause identified was the premature shut down of the descent engines due to a bogus or faulty sensor indication of ground contact. Other probable causes: heat-shield failure, loss of control due to dynamics effects, loss of control due to center of mass offset, landing site not survivable, lander impacted by the backshell or parachute covered the lander (Jet Propulsion Laboratory Special Review Board, 2000).

Figure 16 shows the different possible, impossible, and possible-but-without-substantiating-evidence causes for the MPL failure in relationship to the landing sequence.

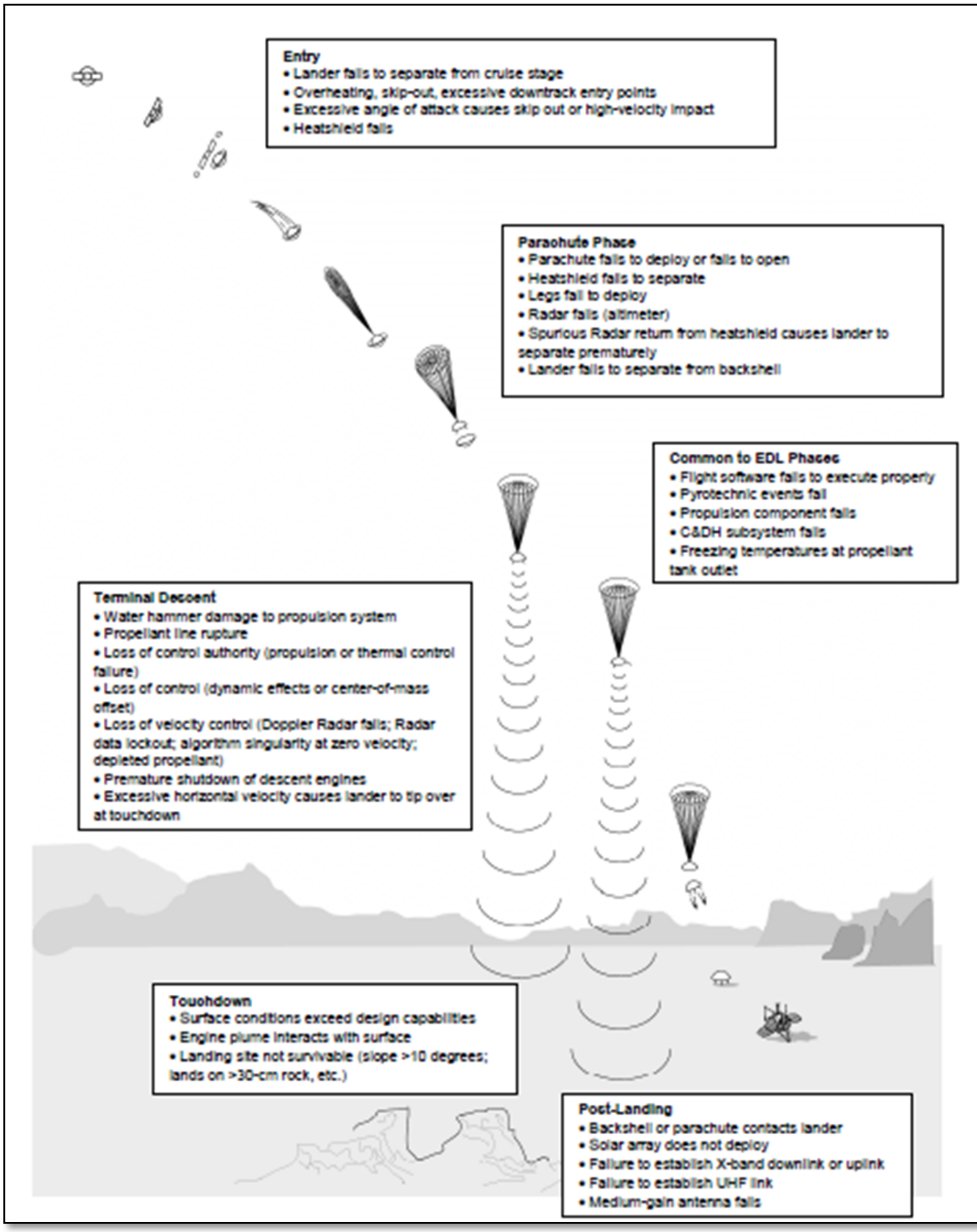


Figure 16. MPL entry, descent, and landing potential failures (From Jet Propulsion Laboratory Special Review Board, 2000)

D. FINDINGS AND RECOMMENDATIONS

The JPL Review Board established the possible technical causes for the MPL failure, but the JPL Board also identified project management and systems engineering issues that led to inadequate technical decisions. They discussed the issues as part of their recommendations.

1. Root Causes: Schedule and Budget Pressure

The report from the JPL Review Board stated that the MPL project was under tight budget and schedule constraints from the beginning. In order to compensate, JPL staffed the project with minimal government support and relied mostly “on Lockheed Martin Astronautics management and engineering structure” (Jet Propulsion Laboratory Special Review Board, 2000, p. 6). In addition, single individuals were tasked with supervising crucial technical areas of the project. JPL employees worked excessive hours (60–80 hours) and they had little time left for project interactions and discussions. For future projects, the JPL Board (2000) stated that systems engineering should be started during the initial phases of the project.

Systems engineering identifies stakeholders and customers’ needs early in the process. This identification helps the prioritization of resources allocation and project goals. Figure 17 shows the different product development stages where the project failed systems engineering-project management integration. The planning and concept development stages are identified, as these are early stages in the process where the stakeholder’s analysis and project goals are identified and refined.

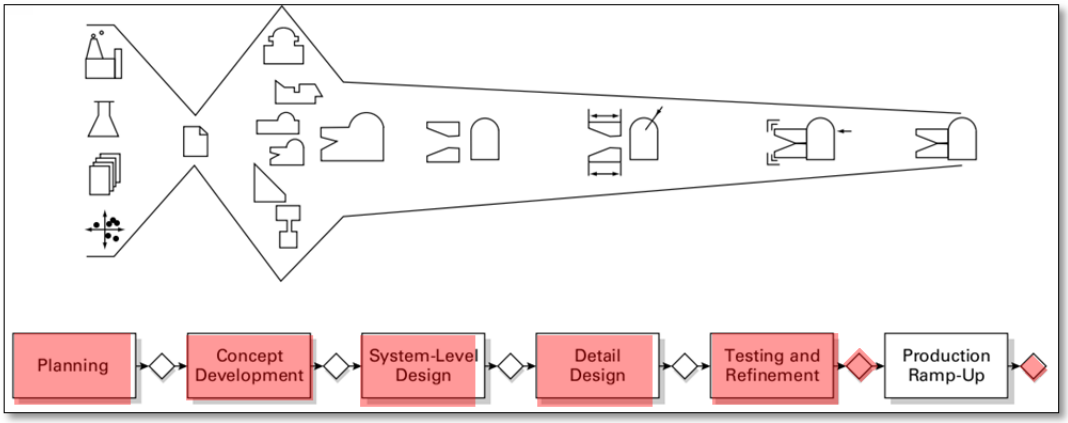


Figure 17. Product development stages where the MPL project showed difficulties (After Ulrich & Eppinger, 2012)

2. Root Causes: Systems Engineering

The JPL Board concluded that "systems engineering resources were insufficient to meet the needs of the project" (2000, p. 9). As a result, some system level analysis and requirements were inadequate or incomplete. Furthermore, DS2 design did not allow for critical mission phases tests to be performed once it was fully assembled.

The final recommendation from the board stated that a project shall maintain adequate systems engineering support throughout the product development process.

E. CONCLUSIONS

The two JPL Review Board recommendations presented above certainly go hand in hand. In *Understanding the Value of Systems Engineering*, Honour (2004) determined, "increasing the level and quality of systems engineering has positive effect on cost compliance, schedule compliance, and subjective quality of the projects." Therefore, inadequate systems engineering will eventually affect schedule and cost.

One of the reasons behind the effect on cost and schedule is that systems engineering guides the clear identification of customer's needs and product requirements. Establishing clear product requirements translates into the test and evaluation parameters needed for systems integration. A clear plan from the beginning could translate in clearer, better, and faster execution of the integration phase.

It is important, therefore, for the systems engineer and the project manager to establish at the beginning of a project a systems engineering plan and a systems engineering management plan to state the level of effort of the systems engineering team. This plan should clearly identify roles and responsibilities and entrance and exit criteria for the predetermined gates (design reviews). Furthermore, establishing this clear entrance and exit criteria in the systems engineering plan helps in the evaluation of the product's technology maturity.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. CASE STUDY 3: DEMONSTRATION OF AUTONOMOUS RENDEZVOUS TECHNOLOGY (DART)

A. INTRODUCTION

This chapter discusses NASA's DART program. A background on the program and the system's failure is presented. The discussion also includes the results of the analysis performed by NASA's Mishap Investigation Board. The later part of the chapter contains an analysis of the findings from a project management-systems engineering interactions perspective. The chapter concludes with a relationship analysis of the issues identified and the Product Development Process.

B. BACKGROUND

DART (Figure 18) was a flight demonstrator intended to conduct autonomous rendezvous maneuvers. DART was envisioned as a leap forward for the United States (U.S.) Space Program:

> Future applications of technologies developed by the DART project will benefit the nation in future space systems development requiring in-space assembly, services, or other autonomous rendezvous operations. (Marshall Space Flight Center, 2004, p. 1)

Orbital Sciences Corporation (OSC) proposed DART in response to a 2001 NASA Research Announcement 8-30 (NRA 8-30) 2nd Generation Reusable Launch Vehicle (2nd GRLV). NASA awarded the DART contract to OSP "as a high-risk technology demonstration project". In November 2002, 2nd GRLV became two new programs: the Orbital Space Plane (OSP) Program and the Next Generation Launch Technology (NGLT) Program. DART became part of OSP. It was at this point that DART gained greater emphasis "because automated rendezvous technology was considered to be critical in supporting the potential future needs of the International Space Station Program" (Marshall Space Flight Center, 2005).

1. Concept of Operations (CONOPS)

DART was developed by Orbital Sciences Corporation of Dulles, Virginia. It was designed to be launched on a Pegasus vehicle using a Stargazer L-1011 aircraft. According to the Marshall Space Flight Center, “Once on orbit, DART would travel around the Earth to rendezvous with the target satellite, the Multiple Paths, Beyond-Line-of-the-Site Communications (MUBLCOM) satellite” (2004, p. 2).

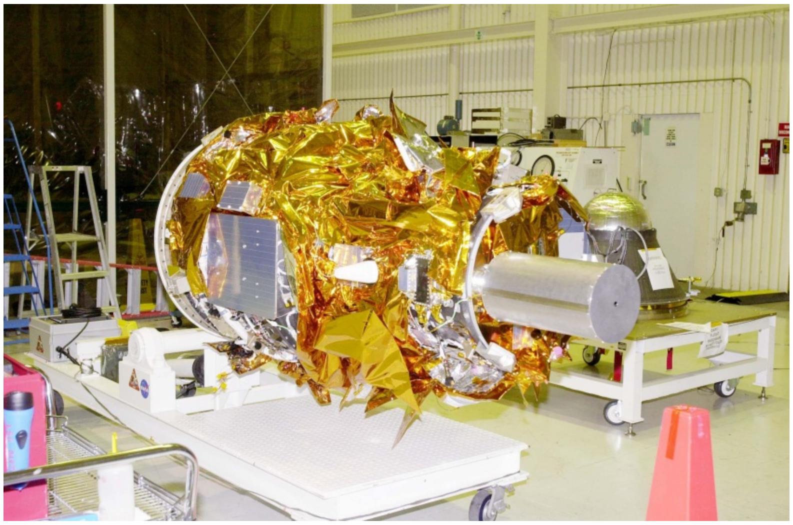


Figure 18. DART at Vanderberg Air Force Base (From Wikipedia DART (satellite), n.d.)

During its mission, DART would demonstrate its advanced video guidance sensor (AVGS). The demonstration of AVGS was key for the whole mission, as it had advanced optics and electronics that would allow DART to communicate with MUBLCOM (Figure 19) and conduct proximity maneuvers “within a range of 5 to 250-plus meters” (Marshall Space Flight Center, 2004, p. 2).

Once DART reached a station keeping position, it would start different rendezvous maneuvers moving closer to and away from the target satellite. It would eventually move away from the target satellite and go into “departure burn (to move it away from MUBLCOM), expel its remaining fuel, and place itself into a short-lifetime retirement orbit in compliance with NASA safety standards” (National Aeronautics and Space Administration, 2006, p. 3).

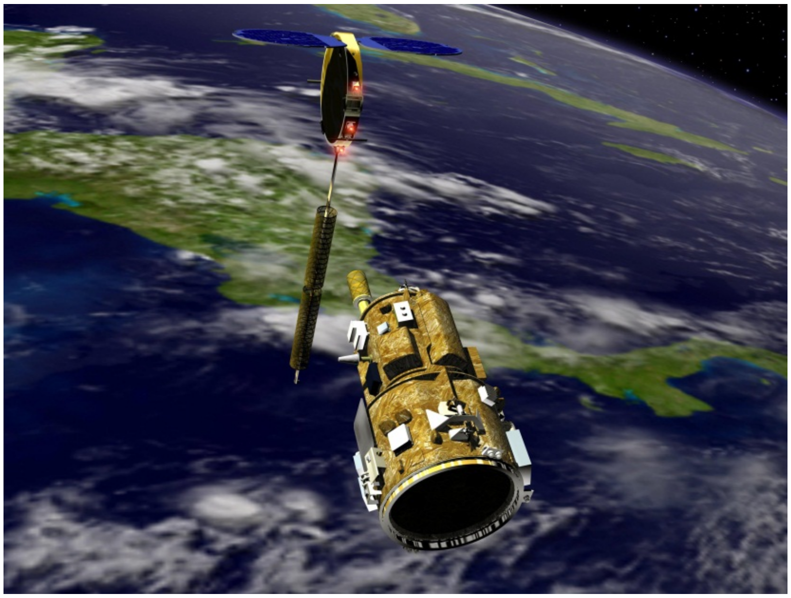


Figure 19. Artist depiction of DART and MUBLCON satellite (From Wikipedia DART (satellite), n.d.)

C. THE MISHAP

The “Overview of the DART Mishap Investigation Results” (National Aeronautics and Space Administration, 2006), indicated that DART was launched effectively from the Pegasus rocket on April 15, 2005.

DART carried out the first phases of the mission successfully. During the later stages (proximity maneuvers) DART started using more fuel than had been anticipated in flight estimates. Eleven hours into the mission, DART determined it had reached low fuel levels and began moving away from the target satellite and initiated a rocket engine burn. In addition to cutting the mission short from its planned original 24 hours, DART bumped into the MUBLCON. The MUBLCON was “pushed into a higher orbit” (National Aeronautics and Space Administration, 2006, p. 4).

D. FINDINGS AND RECOMMENDATIONS FROM THE MISHAP INVESTIGATION BOARD (MIB)

The MIB found four reasons for the DART's premature retirement. All related to software issues:

- Inadequate error between the calculated and measured positions
- Flawed velocity calculations
- A "navigational system overly-sensitive to erroneous data"
- "Incorrect gain control in the calculations"

The MIB further identified the root causes that led to these software problems. The following sections present some of these root causes as well as an analysis of their relationship to the Product Development phases and the systems engineering—project management relationship.

1. Root Causes: System Requirements

Figure 20 shows where in the product development process the different root causes took place in the DART Program.

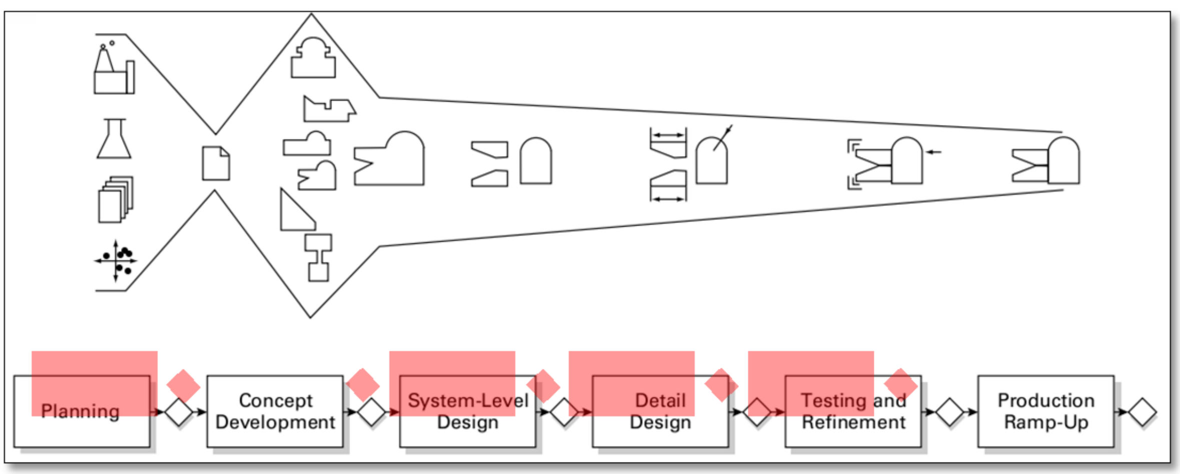


Figure 20. Phases within the DART product development process where issues were identified. (Phases highlighted in red showed issues) (From Ulrich & Eppinger, 2012)

DART was procured under the NRA announcement (as a high-risk low-budget technology demonstration). According to the National Aeronautics and Space Administration Engineering and Safety Center (2006), “NASA procured only the data and set broad requirements.” It was up to OSC to determine how to meet those broad requirements through the detail design. OSC based many of the design aspects on the Pegasus vehicle design. As a consequence, some of the software features, though adequate for Pegasus, where inadequate for “autonomous in-space operations” (Marshall Space Flight Center, 2005).

The MIB recommended that the NRA process be used for “initial conceptual designs.” Mission spacecraft design should be procured under other type contracts with higher levels of scrutiny and government control on systems specification and design features (National Aeronautics and Space Administration, 2006).

2. Root Causes: Inadequate Systems Engineering and Schedule Pressure

Figure 21 shows the stakeholders in the DART program. In an environment where there exist a large number of stakeholders, inadequate management and coordination of the organizations involved in the system design could lead to inadequate system performance. The MIB discovered that a series of design issues were not reviewed or tested adequately due to poor systems engineering and systems integration processes. The MIB recommended that NASA require rigorous training for systems engineers in addition to training project and program managers in systems engineering.

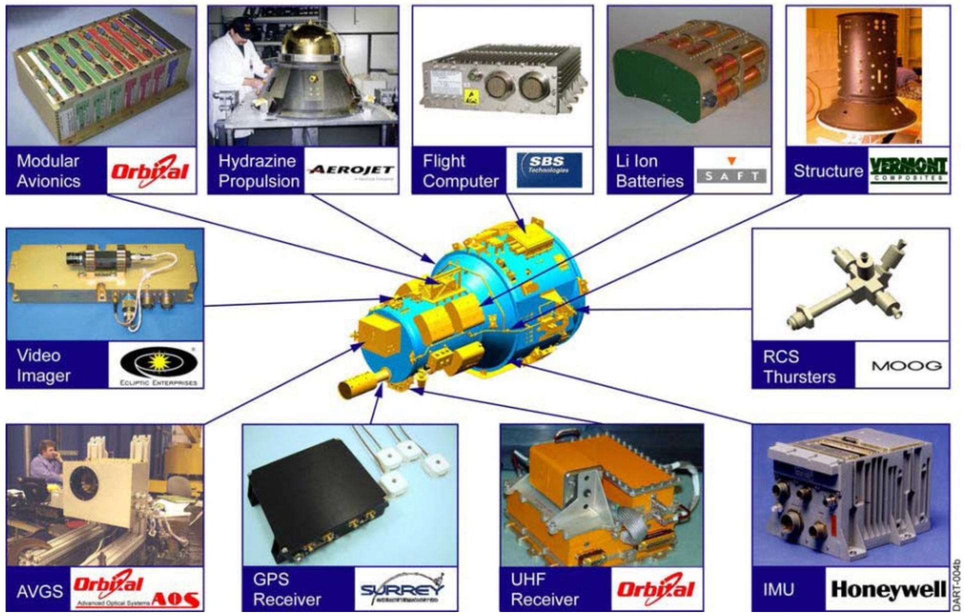


Figure 21. DART's main components and manufacturers (From National Aeronautics and Space Administration, 2010)

3. Root causes: Schedule Pressure and Lack of Adequate Gates for Design Maturity

There was a "late change to the navigations logic's gain setting." Due to schedule constraints, the program continued moving forward without the evaluation of this change through testing. Consequently, the DART team never discovered a problem with a "lower gain setting" that eventually contributed to the mishap (National Aeronautics and Space Administration, 2006).

The MIB recommendation stated the implementation of "checks and balances" throughout the entire development process, from concept design to operational stage, to ensure adequate maturity of the design and technically sound peer review of the efforts (National Aeronautics and Space Administration, 2006).

The same holds true for contractor work review. MIB recommended frequent technical reviews of the efforts and the use of clear entrance and exit criteria prior to each milestone review.

4. Root Causes: Risk Management and the Business Case

DART was originally a “low-cost high-risk demonstration.” The business case evolved and DART became a poster child for NASA’s autonomous vehicles programs. However, DART was still managed under the NRA process and the level of government oversight was not increased and NASA’s systems engineering processes and software design requirements were not enforced.

The MIB report stated, “A rigorous assessment and decision process for managing risk includes ongoing evaluation of NASA’s priorities” (National Aeronautics and Space Administration, 2006). This work concludes that the DART program failed to evaluate NASA’s priorities and failed to adjust execution to implement more rigorous systems integration practices. As a result, risk management was inadequate for the program and led to things like inadequate testing of important systems features like the collision avoidance sub-system.

E. CONCLUSIONS

The DART program suffered from a series of issues related to systems engineering integration into the project management process.

There was inadequate use of systems engineering at the beginning of the process. This inadequate use of systems engineering resulted from what the MIB identified as a lack of experience by the systems engineering team and a lack of understanding by the project management of systems engineering processes.

This work concludes that the project would have benefited from a more rigorous systems engineering process. The use of technical gates or reviews and the enforcement of systems and software design specifications and standards would have helped guide the integration and identify technical risks. Identification of technical risks would have helped the project manager make more informed budget and schedule decisions to meet the changing business case.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CASE STUDY 4: THE CONSTELLATION PROGRAM

A. INTRODUCTION

This chapter discusses NASA's Constellation program. A background on the program is presented, followed by an account of the reasons behind the program demise.

B. BACKGROUND

In accordance with the *Constellation Program: Lessons Learned* report (National Aeronautics and Space Administration, 2011):

> NASA formed the Constellation Program in 2005 to achieve the objectives of maintaining American presence in low-Earth orbit, returning to the moon for purposes of establishing an outpost, and laying the foundation to explore Mars and beyond in the first half of the 21st century.

The Constellation program would also develop a vehicle (named Orion) to replace the space shuttle.

This program was premised on developing an evolutionary capability approach. The initial capability (IC) would focus on the vehicles and ground infrastructure needed to service the International Space Station (ISS) by 2015. This first stage involved the use of a crew launch vehicle, Ares 1, and a crew exploration vehicle, Orion. The second stage, known as the Constellation lunar capability (LC) would add the capability needed to carry lunar missions. The LC required a cargo launch vehicle (Ares V), a lunar lander (Altair), and required spacesuits. The LC also separated the crew from the cargo in order to improve the probability of crew loss from 1/100 for the Space Shuttle to 1/1000 for the new program (Thomas, Hanley, Rhatigan, & Neubek, 2013). Figure 22 shows an artist's rendition of the Orion and the Ares V. Figure 23 shows a depiction of Ares I and Ares V.

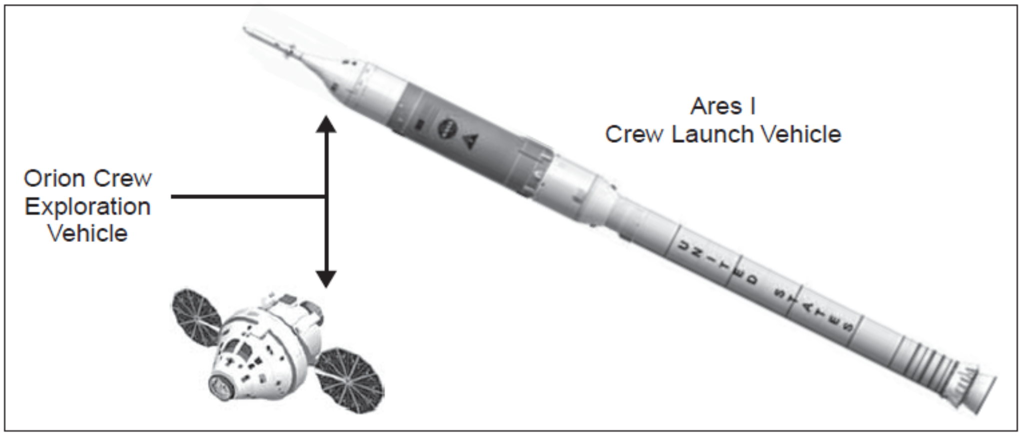


Figure 22. Ares I and the Orion crew exploration vehicle (From United States Government Accountability Office, 2009)

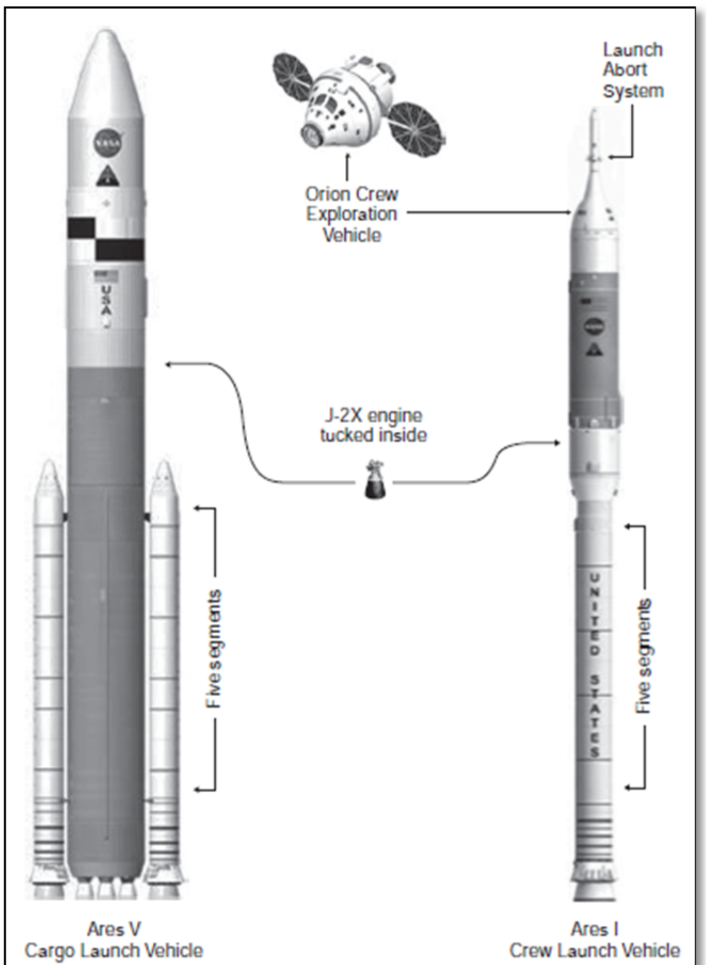


Figure 23. Ares V and Ares I vehicles (From United States Government Accountability Office, 2009)

Johnson Space Center (JSC) managed the Constellation program. The hardware development work spread out through several organizations, as shown in Figure 24.

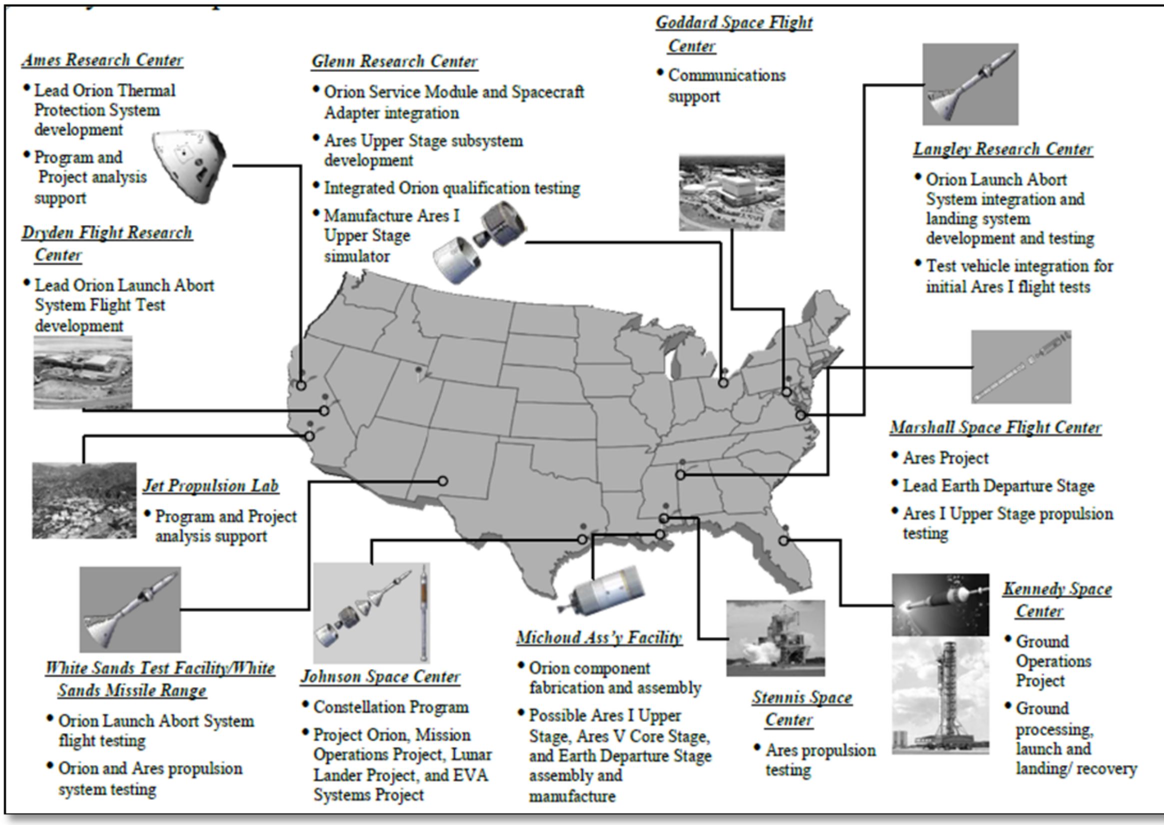


Figure 24. Constellation program allocation of responsibilities (From National Aeronautics and Space Administration, 2011)

C. THE END OF THE CONSTELLATION PROGRAM

In 2009, the Government Accountability Office (GAO) stated that NASA was:

> …still struggling to develop a solid business case—including firm requirements, mature technologies, a knowledge-based acquisition strategy, a realistic cost estimate, and sufficient funding and time—needed to justify moving the Constellation program forward into the implementation phase. (United States Government Accountability Office, 2009, p. 2)

GAO identified an inadequate funding environment. In addition, they identified design and technical risks that might have translated into an inability to meet performance and safety requirements. Because of these issues, NASA had delayed schedule, forward shifting the dates of important milestones. In addition, the lack of adequate funding prevented the NASA team from fully resolving design issues and forced them to shift resources to critical areas that were higher in risk. This shifting of resources resulted in delays in the development of the LC due to allocation of resources to the IC stage and ISS support.

At the time of the GAO report, NASA had allocated $10 million dollars in contracts although there was uncertainty as to the cost of Orion and Ares. Although some features of the Constellation program were kept, the program was cancelled in February 2010 (Thomas, Hanley, Rhatigan, & Neubek, 2013).

D. FINDINGS AND RECOMMENDATIONS

Thomas et al. (2013) identified some key aspects of the Constellation program:

- Scope creep
- Late addition of the system integration function
- Funding uncertainty
- Difficult integration of NADA multi-centers interactions within the program

The next section addresses the first three points.

1. Root Causes: The Late Addition of the System Integration Function and Scope Creep

The Constellation program was a huge, complicated endeavor consisting of several important goals: replacing the Space Shuttle, providing support to the ISS, establishing an outpost on the moon, and eventually transporting humans to Mars. The Orion and Ares tasks, though, preceded the establishment of the Constellation program.

The Orion and the Ares did not have a deep system integration process established. Thomas et al. (2013) stated that the systems engineering and integration process was very lean with only 5–7.5 percent of the budget allocated to it (versus a historical average of 10–15 percent). Integration efforts required for the Constellation program were underestimated based on the IC. Some aspects of the LC were not available at the time of contract elicitation.

According to Thomas et al. (2013), even though the program had a Systems Engineering Master Plan, most of the programmatic decisions (requirements, budgets, design approaches, and acquisition strategies) were developed prior to the systems engineering and integration analysis being completed. Thomas et al. (2013, p. 73), also mentioned the scope creep that resulted from the inability of the team to separate the IC phase requirements from the LC requirements. LC requirements drove many of the IC requirements turning the IC into "more costly and complex than necessary" and increasing the "systems engineering complexity."

Sound systems engineering processes should have preceded major programmatic decisions such as budgets, design approaches, and acquisition strategies. As discussed in the MPL chapter, early implementation of systems engineering facilitates the identification of stakeholders, project goals, and preliminary system level interfaces. Clear identification of stakeholders, project goals and system level interfaces provides the team with the requisite knowledge to identify technical issues, budget uncertainties, and schedule risks when allocating required resources.

Figure 24 identifies the stages of the product development process where the Constellation program issues were evident. The planning phase, the conceptual-development phase, and the system-level-design phase presented management with major systems engineering conflicts for meeting performance requirements within budget and schedule. Due to the cancellation of the program after the preliminary design review (PDR), Figure 25 does not identify the transitions or gates.

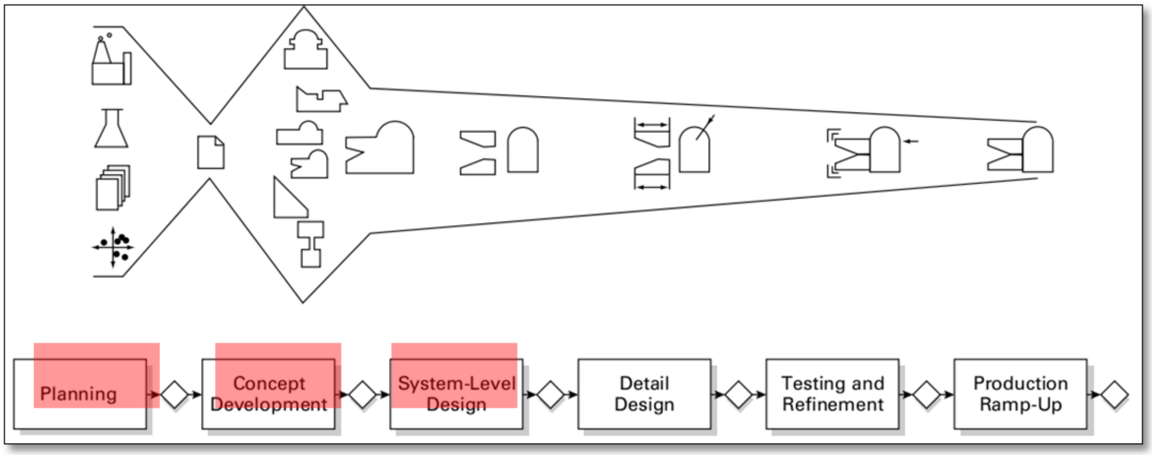


Figure 25. Product development phases where the Constellation program showed conflicts between systems engineering and project management (From Ulrich & Eppinger, 2012)

2. Root causes: Funding Uncertainty

Funding within the Constellation program was uncertain. Thomas et al. (2013) stated that there was a 10 percent cut in budget prior to entering an early major milestone, Preliminary Design Review (PDR). Decisions made during the planning stages, such as commonality between the IC and the LC, which would not be achievable under the new budget realities. In order to meet new budget constraints, management had to select one of two options: delay schedule or down scope the mission. The decision was to delay the schedule.

Yearly continuing resolutions (CRs) also plagued the Constellation program. This uncertainty in funding had a strong, negative influence on the program, as it was difficult to maintain developmental efforts when project personnel were unsure of tasks, paychecks, and government commitment to the program. Figure 26 shows a comparison between a typical development curve and the exiting Constellation program's budget to achieve initial operating capability (IOC). Figure 27 shows the budget cuts to the program through the years. From 2006 through 2010, Constellation received at least five billions dollars less than what was initially estimated the program would need.

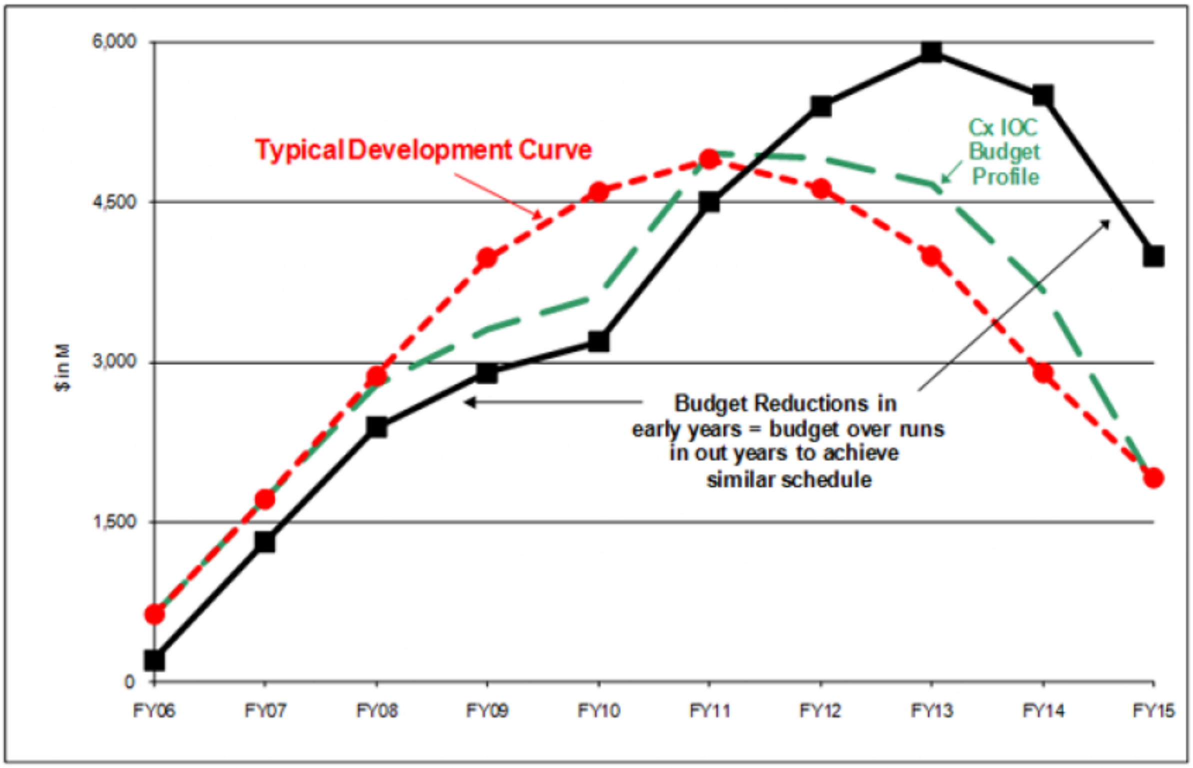
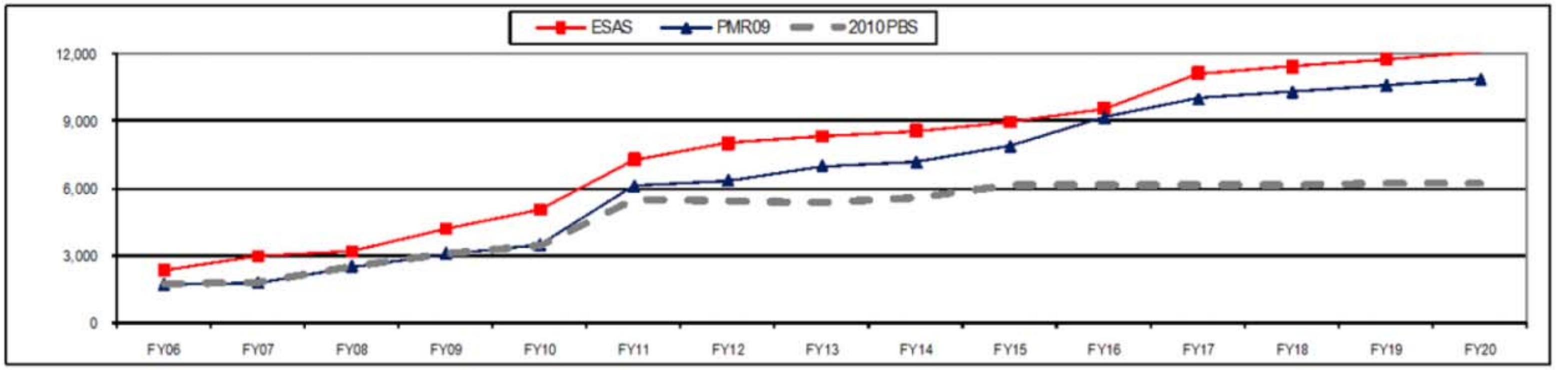


Figure 26. Typical development curve versus Constellation budget profile for IOC (Cx IOC= Constellation initial operating capability) (From Thomas, Hanley, Rhatigan, & Neubek, 2013)



RY$ In M	FY06	FY07	FY08	FY09	FY10	FY11	FY12	FY13	FY14	FY15	FY16	FY17	FY18	FY19	FY20	Total
ESAS	2,333	2,976	3,195	4,202	5,034	7,273	7,994	8,324	8,558	8,948	9,547	11,107	11,422	11,744	12,073	114730
PMR09	1707	1779	2514	3085	3454	6085	6346	6991	7145	7856	9145	9983	10294	10582	10854	97820
2010 PBS	1707	1779	2514	3085	3454	5524	5444	5376	5570	6153	6161	6170	6178	6186	6195	71496
PMR09 vs. ESAS Delta	(626)	(1197)	(681)	(1117)	(1580)	(1188)	(1648)	(1333)	(1413)	(1092)	(402)	(1124)	(1128)	(1162)	(1219)	(16910)
Passback vs. ESAS Delta	(626)	(1197)	(681)	(1117)	(1580)	(1749)	(2550)	(2948)	(2988)	(2795)	(3386)	(4937)	(5244)	(5558)	(5878)	(43234)

Figure 27. Graph of funding costs for the Constellation program. The Exploration System Architecture Study (ESAS) estimated the initial program budget baseline is represented in red. The blue line depicts the program manager's recommendation (PMR). The gray dashed line represents the president's budget submittal (PBS). (From Thomas, Hanley, Rhatigan, & Neubek, 2013)

According to Thomas et al. (2013), the initial budget estimates for the Constellation program provided enough budget reserve. The purpose of the reserve was to fund risk mitigation efforts. However, due to budget uncertainty and large overhead expenditure, NASA eliminated the risk mitigation efforts. “Risks were diligently tracked and reported, but many lingered and, indeed, accrued due to budget limitations” (National Aeronautics and Space Administration, 2011, p. 72). Without the funding required for risk mitigation, the only possible alternatives left to deal with technical challenges were to down scope the program or shift the schedule to future milestone and delivery dates. The program deemed performance expectations more important than meeting schedule and thus delayed milestones.

Schedule delays did not improve the situation for Constellation: new systems integration issues and risks developed. The main reason for these issues and risks was the inability to provide all required components and subsystems at the needed time for integration tests and evaluations. According Thomas et al., “This was the initiating event for a vicious cycle—growing risk placing increasing demand on depleted reserves, which are met by slipping schedule, which in turn increases risk” (2013). Figure 28 shows the slip in schedule of all major milestones on the program. For example, during the ESAS study, it was proposed that IOC would be achievable by late fiscal year (FY) 2012. In 2006, it was estimated that IOC would be achievable by FY 2014. In 2007, there was an apparent optimistic view that IOC would be achieved by late FY 2013. By 2009, it was estimated that IOC would not be achieved prior to mid-FY 2015.

To add to the budget and schedule crisis, NASA had to decide whether to keep and maintain the Space Shuttle infrastructure. The program’s initial plans envisioned that Constellation would utilize the Space Shuttle infrastructure and workforce (NASA 2011). The infrastructure, although not required for IOC, was needed for LC. In order to maintain the capability to support LC, NASA decided to fund the maintenance of facilities and workforce and left the program with reduced budget for hardware acquisition.

The *Risk Management Guide for DoD Acquisition* (2006, p. 1) states “risk is a measure of future uncertainties in achieving program performance goals and objectives within defined cost, schedule and performance constraints.” The systems engineer on a project shall be diligent to identify risks that will affect the final product and communicate them to the team and management. However, the Constellation program environment left very little time and resources for the systems engineers to effect identified risks, as they had no available funding to execute risk mitigation.

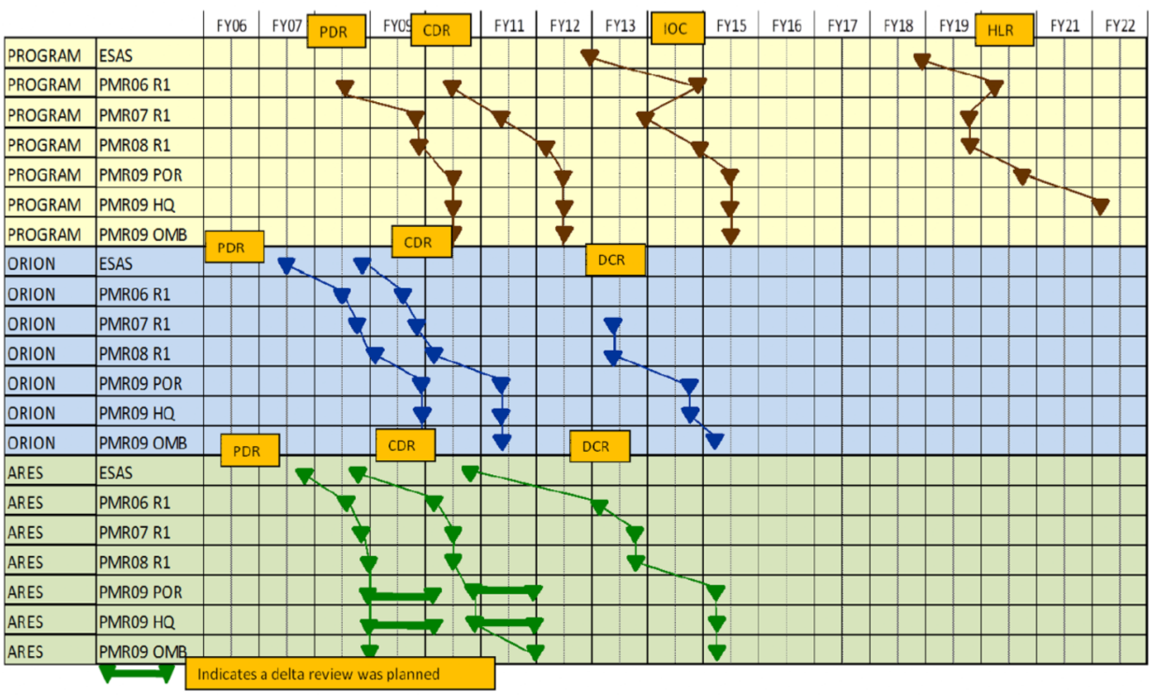


Figure 28. Funding cuts effects on constellation milestones (PDR= preliminary design review, CDR= critical design review, HLR= human lunar return) (From Thomas et al., 2013)

Funding allocation, in this case, became a clear determent to proper systems engineering application. The funding decisions in the Constellation program demonstrate the eternal struggle within a product development process to balance out schedule, budget, and performance. The decisions sacrificed schedule in an attempt to meet the budget constraints. With the early implementation of a systems engineering and integration analysis, the program may have identified design and integration risks earlier. The program would have saved money in re-testing and re-planning that had to occur due to the late incorporation of the systems engineering and integration process.

E. CONCLUSIONS

The Constellation program was a major development and integration endeavor that was subjected to budget cuts and continuing federal funding resolutions. It is important to note though, that some of the decisions made aggravated the situation instead of improving it.

The execution of systems engineering and integration analysis after established programmatic decisions (requirements, budgets, design approaches, and acquisition strategies) resulted in inadequate identification of design and integration risks. Although it may have been difficult to predict if Constellation could have been saved after all the budget cuts and challenges, adequate systems engineering and integration processes would have helped in the analysis of trade-off studies and in the identification of critical technologies that would have produced at least the IC.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. ANALYSIS AND DISCUSSION

A. INTRODUCTION

This chapter summarizes the most common conflicts identified in the previously discussed case studies. A literature review presents the perspective of other researchers on such issues and compares their experiences with the discussed case studies.

B. COMMON ISSUES

Table 1 summarizes the issues discussed in the previous chapters. A few common topics among the four projects are:

- Inadequate systems engineering process or systems engineering started late in the process
    - inadequate product requirements
    - inadequate testing and quality parameters
    - inadequate documentation
- Inadequate risk management
- Insufficient budget and tight schedule

According to Langford (2012) conflicts can be thought of as the interactions between people (objects) that result in an increase in the use of energy, matter, material wealth, or information compared to objects without interactions. A consequence of conflict is the determination of a minimum loss due to the interaction. The reason for conflict is rooted in the difference of interests between two parties. Project managers are prepped to manage budgets and schedules (*of* product development), whereas systems engineers are focused on delivering functions, performance, and quality (*for* products during development). The differences of the interests between project managers and systems engineers for developing products, results in conflict when budgets or schedules are changed from those planned initially and if problems arise that change expectations of stakeholders (such as a product's performance failing to meet requirements).

Table 1. Summary of issues within the programs through the different product development phases

		Programs			
		Hubble	MPL	DART	Constellation
Product development stages	Planning	- Documentation - Inadequate risk management process	- Systems engineering not started from the beginning - Inadequate funding and tight schedule	- Inadequate requirements	- Budget, design approach, and acquisition strategy decided prior to conducting a systems integration analysis
	Concept Development				
	System Level Design	- Lack of adequate interaction among component developers - Inadequate Risk Management Process		- Inadequate systems engineering	- Inadequate risk analysis
	Detail Design				
	Testing and Refinement	- Inadequately staffed QA team developed QA plan that lacked traceability of QA requirements to components and testing requirements.	- Design did not allow for critical mission phases tests to be performed once it was fully assembled		- Inadequate identification of design and integration risks - Schedule changes made it difficult to provide all required hardware for integration tests
	Production Ramp-up				
	Progress Gates	- Inadequate risk management process		-Inadequate implementation of gates - Inadequate Risk Management	

At the highest level of abstraction, a common set of activities can be defined that includes all work carried out by project management and systems engineering. The formalism that outlines this common set of activities builds on the unpublished “Software Project Management Metric—Theoretical Basis, 9 November 2007” by Kadir Demir. The basic structures of work carried out by project management and systems engineering are enacted through defined (and similar) relations between objects and objects, and objects and processes (Langford, 2012).

The basic structures of work are roots for conflict between project management and systems engineering. The following basic structures are redacted from Demir (2007):

- Create—an object can be created as a result of a process
- Delete—an object or process can be deleted
- Transform—an object can be transformed to another object as a result of a process
- Divide—an object or a process can be divided into smaller processes or objects
- Aggregate object—objects can be aggregated into an object
- Aggregate process—processes can be aggregated into a process
- Next and Previous Object—objects can be followed or preceded by other object(s)
- Next and previous process—processes can be followed or preceded by other process(es)
- Requires—an object or process may need another object or process to exist

Conflict is recognized in Table 1 as a result of the work planned or performed in the indicated product development stages.

The next sections provide a literature discussion about the project management and systems engineering relationship and the issues mentioned above.

C. INSUFFICIENT SYSTEMS ENGINEERING IN THE PRODUCT DEVELOPMENT PROCESS

Lilburn (1996) analyzed the process followed by the Lockheed Martin Idaho Technologies team in trying to put together project management training. The objectives of the training were to instruct employees on how to manage project work, implement a systems engineering process and culture, and manage compliance with project management and systems engineering requirements. While developing this training, a problem was encountered when addressing the matter of integrating systems engineering into the way of doing business. Specifically, integration needed to be built into to the work effort and not just something performed on the side.

A functional decomposition was developed to identify the performance related efforts that were essential to “producing products for a customer” (the top-level function) Thus, their approach was focused on the integration of systems engineering and the business process (program management) within a product development process.

Lilburn (1996) questions whether systems engineering is part of planning the customer’s product or if it is part of producing the customer’s product. Through the functional decompositions developed for the training, Lilburn concluded that systems engineering is part of both. Lilburn further concluded that systems engineering is at the heart of the listening process (understanding the customer’s need), the creative process (identifying the product that best fits the need), and the verification process (does the product produce what the customer needs?) Therefore, Lilburn concluded that integrating systems engineering and program management starts with the project team working together to meet the customer’s needs. Primarily, the systems engineer and the project manager shall work together to identify and meet the customer’s need. Perhaps the single most important conclusion from Lilburn’s work is the fact that he identifies that systems engineering is part of the product process from beginning to end (listening process, creative process, and verification process).

Smith, Cowper, and Emes, (2004, p. 9) proposed that starting systems engineering late in the process problem

> …manifests itself as an apparent failure to manage technical risk.…Excessive, early commitment may be at best nugatory, and at worse a blind alley…Late commitment will lead to a lack of competitiveness, failure to meet development schedules or disappointing performance/reliability in the delivered system.

Smith et al. (2004) resonates with Honour's (2004) statement that the quality and level systems engineering efforts will affect cost, schedule, and quality of the project.

The works of Smith et al. and Honour (2004) proposed that systems engineering efforts do have an effect on project cost and schedule compliance and project quality. Later in 2009 through the comparison of successful to less successful programs, Honour (2009, p. 15) stated:

> …poor programs expend comparatively less (systems engineering) effort in the front-end activities (mission definition, requirements engineering) and greater effort in the later, hands-on activities (system design, system integration, and verification/validation).

Lilburn's (1996) work proposed that systems engineering is part of the listening, creative, and verification process. Lilburn's (1996) work also proposes that integrating systems engineering and project management starts with a working together to meet the customer's need. In analyzing Smith et al. Honour (2004, 2009), and Lilburn's (1996), this work concludes that, in order to maximize the cost and schedule benefits to the project and maximize the what Lilburn's (1996) called the listening, creative, and verification processes, systems engineering shall be started in the early stages of the product development process.

The conclusion that systems engineering shall start from the beginning of the project contrasts with the manner in which the case studies presented on this work evolved. Specifically, the Constellation program made programmatic decisions (requirements, budgets, design approaches, and acquisition strategies) prior to completing the systems engineering and integration analysis. The Constellation approach, according to Lilburn (1996), may have skipped the important questions and proceeded to budgeting and design without having the full answers to:

- What is the customer's need (the objective of the project)?
- What product would best fit the need?
- How will the team determine if the product solves the problem or meets the projects objectives?

As a result, the team may have planned a project that could have ended up with insufficient resources. Lack of full understanding of the efforts required to integrate, verify, and validate the needed product will eventually lead to an underfunded project with a tight schedule. This assertion explains what ultimately led to the cancellation of Constellation.

In a similar way, the DART program had little government involvement in the development of requirements elicitation. This lack of involvement showed a lack of sufficient systems engineering in the identification of the customer's needs. In the case of the DART program, this eventually translated into design and integration issues and inadequate assertion of the design maturity during design gates (or reviews).

A poor product requirements development process hinders the evaluation of product performance, thereby making process gates (such as design reviews) ineffective in the assertion of technology maturity. "In the absence of traceable requirements management, requirements remain difficult to verify and systems performance validation is often problematic" (Smith et al., 2004). Therefore, ineffectiveness of the gate resides in a lack of a well-established product performance requirements baseline with which to compare the recommended design.

D. INSUFFICIENT BUDGET AND TIGHT SCHEDULE

Bahill and Briggs (2001) studied the issue of systems engineering started late in the product development process. They called the issue the "systems engineering started in the middle." Also, Bahill and Briggs (2001) discussed that textbook cases present the systems engineering process usually starting at the beginning of the project. They presented the systems engineering process as being involved in the problem identification and the analysis of alternatives. Bahill and Briggs pointed out that real-world systems engineering occurs under different circumstances. In day-to-day projects, systems engineering is started somewhere in the middle. This assertion from Bahill and Briggs is exemplifed the Constellation program scenario.

In addition, Bahill and Briggs identified several reasons for systems engineering to be started in the middle rather than at the beginning:

- lack of experience from management;
- management's impression that systems engineering costs too much;
- management's belief that their process was sufficient because it had not failed in the past;
- management's understanding that they were doing systems engineering but nothing was documented.

Furthermore, Bahill and Briggs (2001) stated that, when a project starts its systems engineering in the middle, the cost is two to ten times as much as a systems engineering started at the beginning of the system life cycle. They concluded that, when starting systems engineering in the middle, a complete systems engineering job cannot be done because it would actually cost too much (Bahill & Briggs, 2001).

Smith et al. (2004) refer to the problem of the "systems engineer in the middle" as "ignoring the left shift." They defined "left shift" as the earlier investment in systems engineering best practices within the development cycle. The problem of ignoring the "left shift," Smith et al. (2004) will become evident during the validation and verification stages, due to the lack of traceable requirements. In consequence, the project may be in danger of an apparent failure to manage technical risk, meet development schedule, and disappoint customers and users with poor system performance or reliability.

In the case of the Hubble telescope, there was an inadequate development of quality assurance requirements. The result was a project with an insufficiently staffed quality assurance group who were unable to provide adequate overview of the quality assurance requirements. This group was even further reduced in size to balance out budget constraints.

Also, the conclusions found in Bahill and Briggs (2001) and Smith et al. (2004) explain why, in part, Constellation may have had serious budget issues. In addition to budget cuts and continuous resolutions, Constellation's inability to start a robust systems engineering process from the beginning may had eventually led to an ever increasing cost to accomplish systems evaluation and integration efforts.

It is therefore understandable that of the four case studies analyzed, three (Hubble, MPL, and Constellation) showed budget pressures and one showed schedule pressure (DART). All the projects showed an inability to implement adequate systems engineering. Incorporating unplanned systems analysis, tests, and integration during later stages of the product development will eventually lead to cost increases and schedule shifts.

E. INADEQUATE RISK MANAGEMENT

Very much in line with Lilburn (1996), Smith and van Gaasbeek (1996) stated that while project managers look at the overall project in terms of cost, time, performance and constraints, the systems engineer's focus is most probably directed more toward the product of that project. They concluded that the project manager's focus is directed on the overall project, whereas the systems engineer's focus is directed toward the product of the project.

When started in the middle, as described by Bahill and Briggs (2001), instead of guiding the product process and then the focus of systems engineering will change to the project process. Instead of identifying the customer, their needs and the right product to address these needs, the systems engineer would focus on the management team as the customer and would attempt to optimize the project process instead of guiding the product process.

Similarly, Considine (1997) stated that the systems engineer focuses on requirements definition for the product and the eventual verification of the product design. The systems engineer's focus, stated Considine, is to ensure the development of an operationally sound system. In this process, previous decisions build the foundation for the history and requirements of the product. These decisions "cannot be ignored … since they may well need to be revisited." In contrast, the project manager would focus on delivery of the process. The project manager would concentrate on maintaining progress to get to the next project milestone.

According to Cosidine (1997), the systems engineer's focus is to ensure the development of an operationally sound system. However, Bahill and Briggs (2001) stated that the systems engineer's focus changes when systems engineering is started in the middle. According to Bahill and Briggs (2001), the focus of the systems engineer changes to optimize the project process. Starting systems engineering in the middle results in a systems engineering process tailored more toward management and not necessarily toward the customer or product. This approach would bring major technical risks to the product that would eventually translate into poor product performance, budget, and schedule overruns. The reason for this risk is that the systems engineer may move into optimizing the project process and design gates instead of optimizing the product.

Shifting the systems engineering focus from the product to the process can greatly affect product requirements definitions. This could happen because the team may be more focused on moving the project along than spending the required time to analyze and correctly define requirements. Maintaining focus on the systems engineering process is essential because, as stated by Meier (2008), systems engineering “ties the technical solution to high-level requirements and maintains the program baseline”.

In his work, Meier (2008) highlights the risks of poor traceability to requirements high-level requirements. Among the risks identified are:

- Developing a technical solution and architecture well before and analysis of alternatives has been conducted,
- Rush into execution phase before facts are in,
- Sense of urgency leads to decisions being made in the midst of inadequate technical, operational, and system understanding.

The above situations, Meier (2008) stated, will eventually lead to unrealistic cost estimates and un-executable schedules. According to Meier (2008), one of the reasons for having unrealistic cost estimates is that operating under any of the circumstances mentioned above will result in changes to requirements and the “addition or modification of requirements almost always leads to cost and schedule growth.”

The above discussion clarifies why three out of the four case studies had risk management issues. Even the MPL, where the investigation board did not identify risk, was identified as having budget and schedule pressure. This budget and schedule pressure led to the conclusion that at some point risk management missed something. In analyzing these four case studies, this work therefore concludes that the inadequate risk management in these projects was probably linked to the faulty definition of the system, its interfaces, and the resources needed for development, test and evaluation, and verification and validation.

Therefore, the discussion leads to the conclusion that, a poorly performed systems engineering process, will result in elevated cost, schedule, and performance risks. This conclusion is similar to the Honour’s (2004) conclusion that “increasing the level and quality of systems engineering has positive effect on cost compliance, schedule compliance, and subjective quality of the projects.”

F. WHAT IS THE SOLUTION?

According to Smith et al. (2004), problems experienced by the implementation of systems engineering are interface failures within the business system. Without substantial investigation and research, systems engineering seems not to be the issue. The issue stated by Smith et al. was “the interface of the systems engineering functions with other elements of a business system model.”

Now this bears on the question of why organizations are failing to implement systems engineering when the majority have a defined systems engineering process. In the case studies presented, the issue was never one of a lack of a systems engineering process. The issue was one of and inadequate integration of the systems engineering process into the project process.

Boardman (1994) stated that poor integration of systems engineering with business processes and other project processes may be a result from different factors. One factor is the failure to understand each other’s processes. In his work, Boardman proposed that carrying out a project involves two issues: getting on with the project and seeing that the “getting on” is proper; well executed; with a sufficiency of process understanding.

Boardman (1994) regarded systems engineering processes as the “getting on” part and the project management as the “seeing to” this “getting on” and concluded that it is important that these two be harmonized. The challenge, he concluded, is “to find a system of shared values which will enable and sustain the correct attitude among engineers, managers, and the other agents within the business.” In other words, it is of primary importance something will unite the team into a common goal.

Laporte (1998) studied the problem of the integration of software engineering, systems engineering, and project management processes. The study found redundancy of efforts among the three processes yet the three were treated differently due to inherent differences in the language and procedures of the disciplines. Risk management was one of the main activities addressed in the three processes. Laporte then proposed the need for common vocabulary and processes that do not contradict each other.

In what could be the solution to the situations exposed by the above discussion, Lilburn (1996) stated that, in order to integrate systems engineering and program management, personnel must be trained and a change in organizational culture was needed. In a similar manner, Mooz and Forsberg (1997) address the culture of systems engineering and project management stating that the "discipline separateness is promoted by universities and the corresponding professional organizations." Mooz and Forsberg conclude that the separation among the disciplines results in project managers that manage cost and schedule and not the technical content while the technical disciplines "ambivalent to the cost and schedule consequences, pursue superior technical solutions."

This work concludes that an optimal integration of systems engineering and project management requires systems engineers trained in project management, and project managers trained in systems engineering. In addition, the common language that Boardman and Laporte allude to could be risk management. Working together, the project manager and the systems engineers should be able to outline the technical and programmatic risks that will help in the budget and schedule management.

G. CONCLUSION

This chapter has discussed a literature review that reveals what might be some of the reasons behind the root causes for the mishaps of the discussed case studies. The chapter concludes that the biggest conflicts in the integration of systems engineering and project management in the product development process are due to:

- insufficient systems engineering in the product development process
- insufficient budget and tight schedule
- inadequate risk management

The chapter concludes by stating that the optimal integration of systems engineering and project management requires systems engineers trained in project management, and project managers trained in systems engineering. In addition, the common language that will help manage the budget and schedule is an acceptable risk management.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSIONS

A. INTRODUCTION

This chapter discusses the general observations, conclusions, and analysis from this work.

B. OBJECTIVES

This work sought the answers to the following questions:

- What are the most common conflicts between program management and systems engineering during product development?
- Where in the product development cycle do they occur?
- How can they be mitigated?

1. What are the Most Common Conflicts Between Program Management and Systems Engineering During Product Development?

This work identified three conflicts in the product development process: insufficient systems engineering in the product development process, insufficient budget and tight schedule, and inadequate risk management. These three situations were found to be the root causes for the mishaps and failures of the case studies presented. Though presented as three reasons, they mainly all arise from either starting systems engineering late in the process or as insufficient application of systems engineering processes in the project as a cost reduction effort.

2. Where in the Product Development Process Do They Occur?

As presented through the discussion of the different case studies, the investigation boards identified issues throughout the product development process. However, in all the cases, it can be stated that failure to establish an adequate systems engineering process in the early planning stages of the product development process will result in issues in the future stages of the process.

Issues presented beyond the planning or concept development stages were mainly inabilities to conduct verification and validation efforts due to poorly elicited requirements or lack of documentation of the requirements elicitation, design, analysis of alternatives, and validation and verification processes.

3. How Can They Be Mitigated?

This work proposes that, in order to mitigate conflicts in the integration of project management and systems engineering, systems engineers and project management shall:

- be able to have a common language
- be able to understand each other's objectives
- be able to understand how these objectives benefit both the product and the project

This work therefore proposes that in order to achieve that common ground and that understanding:

- Systems engineers shall be trained in project management and project managers shall be trained in systems engineering. The cross training should not have the objective of turning systems engineers into project management experts or project managers into systems engineering experts. The objective should be geared towards achieving a true appreciation of each discipline benefits and contributions to the product development process.
- Training should include risk management. Risk management could be the common language between systems engineering and project management. Adequate risk management will help in better allocation of resources, improved budget and schedule management, and better control of overall project scope.

C. FUTURE WORKS

Future works stemming from this work should focus on the identification of the understanding by project managers on the subject of systems engineering and vice versa. In addition, the topic of the relationship of adequate level of effort and quality of systems engineering to effective risk management should be researched and documented.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Bahill, T., & Briggs, C. (2001). The systems engineering started in the middle process. In *Tenth Annual International Symposium of the International Council on Systems Engineering*, Minneapolis, MN.

Boardman, J. (1994). A process model for unifying systems engineering and project management. *Engineering Management Journal*, *4*(1), 25–35.

Considine, V. (1997). Systems engineering and strategic project management. In *IEE Half-Day Colloquium on Systems Engineering in Strategic Managment Planning* (pp. 2/1–2/4). London: Brown and Root.

DART (satellite). (2013, May 27). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/DART_(satellite)

Demir, K. (2007, November 9). "Software project management metric—Theoretical basis" Unpublished interview with G. O. Langford.

Department of Defense. (2011, November 15). Defense Aquisition University. Retrieved June 23, 2013, from https://dap.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=9c591ad6-8f69-49dd-a61d-4096e7b3086c.

Forsberg, K., & Mooz, H. (2002). The case for systems management. In *Proceedings of the Twelfth Annual International Symposium of the International Council on Systems Engineering.* Las Vegas: International Council of Systems Engineers.

Honour, E. (2004). Understanding the value of systems engineering. In *Proceeding of the fourteenth annual International Symposium of the International Council on Systems Engineering.* Toulouse: International Coucil of Systems Engineering.

Honour, E. (2009). Demographics in measuring systems engineering return on investment (SE-ROI). In *Proceedings of the 19th Annual International Symposium of the International Council of Systems Engineering.* Singapore: International Council of Systems Engineering.

International Council of Systems Engineering. (2011). *INCOSE systems engineering handbook v. 3.2.2.* San Diego: author.

Jet Propulsion Laboratory Special Review Board. (2000). *Report on theloss of the Mars Polar Lander and Deep Space 2 missions* (JPL D-18709). Pasadena, CA: Jet Propulsion Laboratory California Institute of Technology.

Langford, G. O. (2012). *Engineering systems integration: Theory, metrics, and method*.Boca Raton, FL: CRS Press.

Laporte, C. (1998). Development and integration issues about software engineering, systems engineering, and project management processes. In *Software Process Improvement.* Software Process Conference, Monte Carlo.

Lilburn, B. (1996). Integrating systems engineering and program management. In *Sixth Annual International Symposium of the International Council on Systems Engineering.* Boston, MA: International Council of Systems Engineering.

Marshall Space Flight Center. (2004). *NASA facts.* Hunstville, AL: National Aeronautics and Space Administration.

Marshall Space Flight Center. (2005). *Demonstration of Autonomous Rendezvous Technology Mishap Review Board.* Hunstville, AL: author

Meier, S. (2008). Best project management and systems engineering practices in the preacquisition phase for federal intelligence and defense agencies. In *Project Management Journal*, *39* (1), 59–71.

Mooz, H., & Forsberg, K. (1997). Visualizing systems wngineering and project management as an integrated process. In *Proceedings of the Seventh Annual International Symposium of the International Council on Systems Engineering .* Los Angeles: International Council on Systems Engineering.

NASA Jet Propulsion Laboratory. (n.d.). *Mars Polar Lander*. Retrieved August 14, 2013, from NASA Jet Propulsion Laboratory: http://mars.jpl.nasa.gov/msp98/science.html

National Aeronautics and Space Administration. (1990). *The Hubble Space telescope optical systems failure report; NASA-TM-103443.* Pasadena, CA: Jet Propulsion Laboratory.

National Aeronautics and Space Administration. (1999). *Mars Polar Lander/ Deep Space 2—Press kit.* Pasadena, CA: author.

National Aeronautics and Space Administration. (2006). *Overview of the DART mishap investigation results.* Retrieved from http://www.nasa.gov/pdf/148072main_DART_mishap_overview.pdf

National Aeronautics and Space Administration. (2010). *The DART mission: hard decisions in a changing environment.* Greenbelt: author.

National Aeronautics and Space Administration. (2011). *Constellation program lessons learned volume 2: detailed lessons learned.* Houston: Lyndon B. Johnson Space Center.

National Aeronautics and Space Administration. (2011). *Constellation program: lessons learned volume 1.* Houston: Lyndon B. Johnson Space Center.

National Aeronautics and Space Administration Engineering and Safety Center. (2006). *NESC review of Demonstration of Autonomous Rendezvous Technology (DART) Mission Mishap Investigation Board Review (MIB).* Hampton, VA: author

Project Management Institute, Inc. (2013). *A Guide to the Project Management Body of Knowledge (PMBOK(R) GUIDE).* Newtown Square, PA: Project Management Institute, Inc.

Smith, A., Cowper, D., & Emes, M. (2004). Systems engineering mangement. In *14th Annual International Symposium of the International Council of Systems Engineering.* Toulouse: International Council of Systems Engineers.

Smith, B., & van Gaasbeek, J. (1996). Project management, systems engineering and the amazing DNA double helix. In *Sixth Annual Internaltion Symposium of the International Council of Systems Engineering.* Boston, MA: International Council of Systems Engineering.

Thomas, L. D., Hanley, J. M., Rhatigan, J. L., & Neubek, D. (2013). NASA's Constellation program: The final word. *Systems Engineering*, 71–86.

Ulrich, K., & Eppinger, S. (2012). *Product design and development* (5^{th} ed.). New York: McGraw-Hill/Irwin.

United States Department of Defense. (2006, August). *Risk managment for DoD acquisition*. Washington, DC: author.

United States Government Accountability Office. (2009). *NASA Constellation program cost and schedule will remain uncertain until a sound business case is established.* Washington, DC: author.

Waldrop, M. M. (1990, august 17). The case of the single-point failure. *Science Magazine*, *249* (4970), 735–736.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California